AFRL-RH-WP-TR-2021-0088

FOREIGN LANGUAGE AUTOMATED INFORMATION RETRIEVAL (FLAIR) / MACHINE TRANSLATION FOR ENGLISH RETRIEVAL OF INFORMATION IN ANY LANGUAGE (MATERIAL)

John Makhoul / Rich Schwartz / Damianos Karakos
Le Zhang / William Hartmann / Manaj Srivastava
Sanjay Krishna Gouda / Lee Tarlin / David Akodes
Raytheon BBN Technologies
10 Moulton Street
Cambridge, MA 02138

December 2021

Final Report

Distribution A. Approved for public release; distribution unlimited.

AIR FORCE RESEARCH LABORATORY
711TH HUMAN PERFORMANCE WING
AIRMAN SYSTEMS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RH-WP-TR-2021-0088 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

ANDERSON.TIMOTHY.RAY.1230210728 Digitally signed by ANDERSON.TIMOTHY.RAY.1230210728 Date: 2022.03.21 09:25:15 -04'00'

TIMOTHY R. ANDERSON, DR-IV, Ph.D.
Work Unit Manager
Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

MURDOCK.WILLIAM.P.1048742161 Digitally signed by MURDOCK.WILLIAM.P.1048742161 Date: 2022.03.21 12:28:26 -04'00'

WILLIAM P. MURDOCK, DR-IV, Ph.D.
Chief, Mission Analytics Branch
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

CARTER.LOUISE.ANN.1230249128 Digitally signed by CARTER.LOUISE.ANN.1230249128 Date: 2022.03.25 13:37:20 -04'00'

LOUISE A. CARTER, DR-IV, Ph.D.
Chief, Warfighter Interactions and Readiness Division
Airman Systems Directorate
711th Human Performance Wing
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
12/22/2021	Final	09/21/2017	12/22/2021

4. TITLE AND SUBTITLE

Foreign Language Automated Information Retrieval (FLAIR)/Machine Translation For English Retrieval Of Information In Any Language (MATERIAL)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8650-17-C-9118		
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		H0V2

6. AUTHOR(S)

John Makhoul, Le Zhang, Sanjay Krishna Gouda
Rich Schwartz, William Hartmann, Lee Tarlin
Damianos Karakos, Manaj Srivastava, David Akodes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Raytheon BBN Technologies 10 Moulton Street Cambridge, MA 02138	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory 711th Human Performance Wing Airman Systems Directorate Warfighter Interactions and Readiness Division Wright-Patterson Air Force Base, OH 45433		AFRL-RH-WP-TR-2021-0088

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

AFRL-2022-0990; Cleared 2 Mar 2022

14. ABSTRACT

This is the final report for the FLAIR team under the IARPA MATERIAL program. The objective of the program was to develop the technologies for cross-lingual information retrieval from speech or text documents in foreign languages. The retrieved relevant documents, based on an English query, are then summarized in English for presentation to the analyst. Progress was made in the four requisite technologies: Automatic Speech Recognition (ASR), Machine Translation (MT), Cross-Lingual Information Retrieval (CLIR), and Summarization. The report summarizes the advances made in the four technologies throughout the program. The project resulted in a number of deliveries of software and data to the Government. The MT and CLIR technologies have been transitioned to the Raytheon BBN Multi-Media Monitoring System (M3S) for eventual deployment in Government installations

15. SUBJECT TERMS

Cross-lingual information retrieval (CLIR), machine translation (MT), automatic speech recognition (ASR), summarization, morphology, statistical modeling, machine learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	45

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Timothy R. Anderson, Ph.D.	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Raytheon BBN Technologies (BBN) is pleased to submit this final report for its work during the Intelligence Advanced Research Projects Activity (IARPA) Machine Translation for English Retrieval of Information in Any Language (MATERIAL) Program. The objective of the program was to develop the technologies for cross-lingual information retrieval from speech or text documents in foreign languages. The retrieved relevant documents, based on an English query, are then summarized for presentation to the analyst. The summary of our work is given here for each of the component technologies.

1.1 Automatic Speech Recognition (ASR)

Our ASR system was a hybrid model with a convolutional neural network-long short-term memory (CNN-LSTM) acoustic model and an n-gram language model (LM). The BBN ASR system has consistently been a top performer across the three phases of the program. Key components include multilingual acoustic model pre-training, semi-supervised domain adaptation, automatic lexicon and LM expansion, recurrent neural network language model (RNN-LM) rescoring, and lattice boosting. The system is also highly efficient, giving us the ability to deliver real-time systems with only a small degradation in performance.

1.2 Machine Translation (MT)

We developed an advanced Neural MT pipeline with bi-directional translation, multilingual training, a per-dataset weighting scheme, and semi-supervised training (SST) using automatically acquired monolingual data. BBN also designed a suite of biased MT techniques, built around a novel lattice boosting framework that enabled superior Summarization output with very high query word coverage.

1.3 Cross-lingual Information Retrieval (CLIR)

Our CLIR component was primarily based on a rigorous probabilistic framework, resulting in strong year-after-year performance in official evaluations. We introduced several techniques that were later adopted by other teams, including the probability-of-occurrence model to match the relevance criterion, taking input from the ASR confusion networks (not just the 1-best transcription), and the Neural Network Lexical Translation Model (NNLTM) for generating *contextualized* lexical translations.

1.4 Summarization

Our summarization system selects one or more snippets from MT output using a submodular selection algorithm that chooses the highest scoring snippets with an emphasis on diversity to maximize coverage of the query words. We modeled the probability that judges accept or reject documents as a function of whether the document was relevant and whether the summary contained the query words. With the use of various MT biasing techniques, we then controlled the probability that the summary contained the query words based on the probability of relevance provided by CLIR.

1.5 Tech Transition

BBN's Multi-Media Monitoring System (M3S), which enables access to video, radio, web, and social media in 22 different strategic languages in real time, was improved by adopting two MATERIAL technologies: Neural MT (NMT) and advanced CLIR. The NMT resulted in significantly improved MT quality. The probabilistic CLIR increased recall by virtue of searching for all likely translations of the English query words in the foreign language.

1.6 Deliveries

We delivered to Massachusetts Institute of Technology (MIT) Lincoln Laboratory: ASR decoding (Tagalog), MT decoding with fine-tuning capability, BBN WebText Collection System, and a Farsi End-to-End (E2E) MATERIAL system as Docker containers. We also delivered monolingual webtext downloaded for all Option Period 2 (OP2) languages plus Pashto.

2.0 INTRODUCTION

This section presents a summary of the changes in our various technologies from the Base Period to Option Period 1 (OP1) and then to OP2.

2.1 ASR

Base Period: The major components of our ASR system were stable across the three phases of the program. Our acoustic models (AM) were deep neural network-based models using convolutional and recurrent layers. Our models were pre-trained on a large set of multilingual data. N-gram language models, built from automatically collected web data, were used during decoding. These technologies served at the basis of our systems, but additional capabilities were investigated in later phases.

OP1 Period: Starting with OP1, we began adding RNN-LM rescoring to all of our systems. After a first pass decode with an n-gram LM, the LM scores are interpolated with the scores from an RNN-LM. We also experimented with incorporating untranscribed data from YouTube in our SST pipeline. The data was downloaded from YouTube from channels that purported to be predominately in the language of interest. The data were initially filtered with language identification and then decoded based on our current best system. A second stage of filtering only kept the data our model decoded most confidently.

In addition to confidence-based selection, we also experimented with two other techniques. The first was i-vector similarity. We computed an average i-vector for the Dev set as a target. This average i-vector was then compared with an i-vector from each YouTube audio file. The audio files that were most similar in terms of cosine similarity were selected. The second technique used a deep neural network for classification. An i-vector was generated for each utterance in the Dev and Training sets. Data from the Dev set were considered in-domain and data from the Training set were considered out-of-domain. The model was trained to distinguish the two sets. Using this model, we selected the utterances from the YouTube data that were classified as in-domain. Neither of the two techniques outperformed confidence-based selection, so we did not use them.

OP2 Period: At the end of OP1 and the start of OP2, we started adding additional transcribed data when it was available. For Pashto, we had broadcast data available at BBN. For Kazakh, we found a corpus of read speech. Both corpora significantly improved the performance of our system on broadcast speech, but had no effect in the conversational domain. We also added an additional post-processing step to our standard pipeline. We boosted the scores of probable query translations in our lattices to improve CLIR. Boosting the scores gave the words a higher likelihood and made them less likely to be filtered out during Consensus Net (CNet) creation. By the end of OP2, we added the ability to combine ASR systems at the CNet level. Since the outputs from multiple systems were combined into a single CNet, only a single search was needed during CLIR. This simplified the process compared to performing a search for each system and then aggregating the results.

2.2 MT

Base Period: We started with a baseline Statistical MT (SMT) system using phrase-based hierarchical MT (PBMT) with string-to-dependency translation rules. The SMT model was augmented with rich features and a Neural Network Joint Model (NNJM) that models a joint target LM and translation model. At the same time, we experimented with Transformer-based Neural MT, which gives more fluent translation and is better suited for Summarization. We observed that, when using only the training data provided by MATERIAL, the SMT was still better than Neural MT. But when using additional data from PanLex and the Defense Advanced Research Projects Agency (DARPA) Low Resource Languages for Emergent Incidents (LORELEI) Program, the NMT outperformed SMT. We also experimented with different morphological segmentations.

OP1 Period: We switched to full NMT model with bi-directional translation, ensemble decoding, multilingual training, an unsupervised tokenizer, and back translation. We addressed the data imbalance issue in multilingual training using a combination of oversampling and a per-dataset weighting scheme. In addition, we developed an SST method using large amounts of WebText automatically downloaded from the Web. As a result of all these improvements, our OP1 MT models gained an average of 4.7 BiLingual Evaluation Understudy (BLEU) points on the Analysis sets of each language over the NMT models built during the Base period.

BBN designed a novel hybrid subword/word MT for Summarization that enabled us to influence query coverage. With biased decoding, we significantly improved Pashto query coverage on the Analysis set from 58% to 91%, which greatly improved the E2E performance of our OP1 Pashto Evaluation (Eval) system.

OP2 Period: We focused on improving data quality in this period and developed a Translation Edit Rate (TER) based filtering method to improve MT by discarding questionable parallel data. For each OP2 language, we downloaded and filtered millions of words of news data in the source language and in English and used them in SST training. Moreover, we improved Kazakh MT by pivoting through Russian using English-Russian and Russian-Kazakh parallel corpora.

We improved biased MT with dynamic boosting that boosts selected query words after CLIR. We also implemented NMT lattice generation, which allows us to further improve query word coverage using Finite State Transducer (FST) lattice biasing. These innovations allowed us to present a diverse array of MT outputs to Summarization, each operating at a different level of query word coverage. Our biased MT systems as a whole ensured that more than 95% of query words were present in the output. This gave the Summarization component the flexibility to choose the best snippet to present to the user.

2.3 CLIR

Base Period: The CLIR technology developed during the base period was mainly in the areas of retrieval modeling and domain modeling. Specifically, we implemented a bag-of-words retrieval model, which follows the BBN probabilistic cross-lingual retrieval technique that was used successfully in the Text REtrieval Conference (TREC) 2001 evaluation. We also implemented a “probability of occurrence” retrieval model, which is more in line with the MATERIAL relevance criterion (at least one occurrence of the query in the document).

For CLIR over speech documents, we implemented a method that performs retrieval over a rich ASR hypothesis space (ASR consensus networks) and works much better than retrieval over the 1-best automatic transcription of the speech utterance.

Since the base period of the program included a "domain" component (both in terms of retrieving documents for domain-constrained queries and in terms of retrieving documents that match a domain), we implemented a weakly-supervised topic classification system using a large English corpus (Wikipedia) for training. The labels for the training of the domain classifier were obtained by using manually-chosen "seed" words, using the IARPA-provided domain description document for guidance.

For deciding what documents to give to the analyst in the output, so that the official performance measure of the program (which tradeoffs misses and false alarms) is optimized, we implemented a principled score normalization and thresholding scheme, based on decision theory.

OP1 Period: The innovations developed in this period had a dramatic impact on the final performance. Specifically, we designed and implemented the NNLTM which allows us to come up with *contextualized* lexical translations of foreign words into query-language (English) words. This model complements the *context-independent* translation probabilities obtained through standard statistical approaches, resulting in large gains in performance.

To handle the complex nature of the MATERIAL queries, we developed a novel *hierarchical* retrieval model that uses the parse tree of the queries as a bottom-up retrieval tree. This hierarchical retrieval mechanism allows us to be more selective and more precise when retrieving documents for two specific types of queries: *phrase queries* (multi-word queries, usually noun phrases) and *conceptual queries* (queries expressing an information need about the "topic" in the document).

In addition to context-independent and contextualized lexical translations, we also use diverse MT outputs during indexing: (i) MT 1-best output; (ii) MT lattices; (iii) biased MT lattices, where paths through the lattice that contain the query words are boosted so that they become more easily detected.

OP2 Period: During this phase, we focused more on data augmentation, computational efficiency, as well as on better thresholding for speech CLIR.

Instead of doing a computationally expensive grid-based search for finding the best set of weights with which different data sources are combined for training lexical translation models, we developed a likelihood-based approach using the efficient Expectation-Maximization algorithm.

We implemented efficient pruning techniques for speeding up retrieval of conceptual queries. We focused on queries with named entities (for which we do not do query expansion) and queries with multiple components (for which the non-conceptual component is used to down-select the set of candidate documents).

In terms of data augmentation, we (i) implemented a transliteration mechanism for handling out-of-vocabulary (OOV) query terms; (ii) increased the amount of training data in a semi-supervised way by pivoting through a third language, and (iii) emphasized parallel sentences that contain the query words by performing information retrieval (IR) on the training data.

In speech CLIR, where the number of queries with relevant documents is much lower than that of Text, we introduced a Gaussian mixture model and a reduced acceptance fraction mechanism as a way to improve the baseline thresholding mechanism, recovering several percentage points in performance.

2.4 Summarization

Base Period: Our Summarization system used a submodular selection algorithm that employed fixed-length snippets around word-level evidence. The system was provided two types of word-level evidence:

- Word-level evidence for query relevance produced by a single CLIR system
- Word-level evidence for domain relevance produced by a Domain ID model

The word-level evidences were provided in source (foreign language) sentences and were mapped to English sentences using a source-to-target alignment model. The English sentences used for creating summaries were output by a single MT system. Summaries were long (around ten snippets, each containing around ten words) and the evidence words were highlighted.

OP1 Period: In this period, we still used submodular selection algorithm with fixed-length snippets around word-level evidence. However, only one type of word-level evidence was used:

- Word-level evidence for query-relevance produced by multiple CLIR systems (domain-relevance was done away with after the base phase)

The system aggregated the word-level evidence provided by multiple CLIR systems. Source-to-target alignment was done away with; query words were directly looked up in the English sentences. English sentences came from multiple MT systems (submodular selection algorithm would use the entire set of English sentences to do the selection from). For earlier languages (Lithuania, Bulgarian), three MT systems (2NMT+1PBMT) were used. However, for the final language (Pashto), only one MT system was used since we did not see any deterioration in Actual Query Weighted Value (AQWV) with a single MT system. This single MT system used lattice boosting. Query words highlighted in the summary were accompanied with footnotes containing multiple possible translations of the highlighted query word in question. These translations came from an interpolation of GIZA (a bitext alignment method for MT) and NNLTM translation models. Later on (for Pashto), an interpolation of NNLTM translations and nearest Global Vectors for Representation (GloVe) embedding neighbors (Pennington et al., 2014) was used. Summaries were generally no longer than two snippets, and these snippets were generally around 20 words long. A “System Confidence” header was introduced at the top of the summary. This captured a measure of confidence (in percentage) that the CLIR system had in the relevance of a given document.

OP2 Period: The summarization system used multiple MT systems, but all these systems were biased (to one degree or another) to contain the query words. Footnotes were done away with since we had not seen much utility for footnotes after ensuring that the summary text had the query word. We still used evidence from multiple CLIR systems, but we ditched word-level evidence for just the sentence-level evidence. For Farsi, we used a combination of lattice-boosted and Sockeye MT to increase the query coverage. For Kazakh and Georgian, we used five MT systems with various levels of biasing along with our AQWV optimization algorithm which selected one MT version with the appropriate level of biasing given the CLIR score of the document.

3.0 TECHNICAL APPROACHES

3.1 ASR

3.1.1 System Description

Our hybrid systems use a Deep Neural Network-Hidden Markov Model (DNN-HMM) approach with the Kaldi toolkit (Povey, et al., 2011), trained using BBN's Sage system (Hsiao, et al., 2016). The input features used in the hybrid model are 40-dimensional mel-filter cepstrum coefficients (MFCC) features, along with a 100-dimension ivector. The HMM structure is the *chain* structure used in (Povey, et al., 2016), which defines a two-state model with a single self-loop, allowing the model to be traversed in a single frame. The acoustic model generates output at a reduced frame rate of 33Hz, one prediction for every three frames.

The neural network used employs a combination of convolution, time-delay, and recurrent layers. The basic structure of the network is similar to the Switchboard setup in (Cheng, et al., 2017). The difference is that we prepend an additional eight convolution layers. The MFCC features are transformed back into log mel-filterbank features and passed through these convolutions in parallel with the first three Time Delay Neural Network (TDNN) layers (which process both the MFCC and ivector features). The output of both of these subnetworks is concatenated before passing through a Rectified Linear Unit (ReLU) layer and continuing to the alternating LSTM/TDNN portion of the network as described in (Cheng, et al., 2017). The total number of parameters varies slightly per language but is approximately 24M.

The hybrid model is initialized from a model trained on 1560 hours of multilingual data from 11 languages. The same data is also used for training the ivector extractor. The model is fine tuned to the target language using the lattice-free maximum mutual information (LF-MMI) objective function (Povey, et al., 2016) for one epoch. It is then trained with state-level minimum Bayes risk (sMBR) (Veselỳ, Ghoshal, Burget, & Povey, 2013) for an additional epoch. The supervised-only hybrid acoustic model and the semi-supervised hybrid model are both trained in the same way with the same model structure; the only difference is the data used.

All of our hybrid systems use a trigram language model. Separate models are built for each language model data source and the final model is an interpolation of the individual models. After an initial decode, the lattice is rescored using an RNN-LM language model. The neural model consists of two LSTM layers and three fully connected layers.

For all languages an initial acoustic model is trained using only the supervised conversational telephone speech (CTS) data. The acoustic model is a multilingual CNN-LSTM. A separate trigram language model is trained for each data source and then interpolated to produce a single model. Decoding uses the expanded lexicon.

We use the Eval set as unsupervised data for SST. The entire Eval set is decoded with the initial model. The AM is then retrained. During LF-MMI training, the supervised data is combined with the automatically transcribed Eval data. Only the supervised data is used during sMBR training.

3.1.2 Performance Improvements over Baseline System

Each step in the ASR pipeline produces an improvement over the previous. Results in Table 1. Baseline ASR WER Performance Across Languages.

show the reduction in Word Error Rate (WER) from expanding the language model, semi-supervised training, and RNN-LM rescoring for each language. Note that the results are generated using a similar pipeline for all languages and is not as good as our final evaluation systems.

Table 1. Baseline ASR WER Performance Across Languages.

System Description	Swa.	Tag.	Som.	Lit.	Bul.	Pas.	Kaz.	Geo.
Baseline Model	45.0	---	---	47.9	40.0	42.8	51.1	39.9
+ expanded lexicon/LM	34.3	35.9	51.8	23.5	22.0	39.9	25.9	25.6
+ SST	31.0	30.9	48.6	19.8	19.1	39.2	17.9	21.4
+ RNN-LM rescoring	30.0	30.0	49.2	18.1	17.7	39.3	16.7	20.2

3.1.3 Expanded Lexicon vs. Expanded LM

Based on previous experience in the IARPA Babel program, we expected ASR performance to improve more by lexicon expansion than by LM expansion. Lowering the OOV rate was more important than adding additional LM data. Results in the MATERIAL program tell a different story, as we show below.

In the last years of the Babel program (where both the acoustic training and test were CTS), we used much less transcribed acoustic training data than in MATERIAL (three or ten hours, depending on the language, vs. 50 to 100 hours for MATERIAL). So, there was a severe OOV problem in Babel and increasing the lexicon using web data had a large benefit. However, there was little benefit for including extra text in the LM because the web sources did not match the style of language in CTS.

Table 2 shows the results of increasing the lexicon and updating the language model using additional text data in MATERIAL. The results were averaged over Swahili, Tagalog, Somali, Lithuanian, and Bulgarian. For Broadcast test, we get a significant gain by expanding the lexicon, as one would expect, but a bigger gain from the LM expansion because the web text data were closer in style to the Broadcast test. In contrast, for CTS test, the vocabulary and LM expansions had little effect because there was already enough training data for the lexicon, and the web data were not a good match to the CTS test.

Table 2. Comparing WER Reduction from Expanding Lexicon and LM.

Model Description	Broadcast	CTS
Baseline Supervised Model	49.4	37.5
+ Lexicon Expansion	44.4	36.8
+ LM Expansion using bitext only	35.8	37.3
+ LM Expansion using web data	31.5	36.2

3.1.4 Additional Transcribed Acoustic Data

For two of the languages in the IARPA MATERIAL program, we had access to additional transcribed data. BBN had previously transcribed about 35 hours of broadcast speech. The data were released to all MATERIAL participants. For Kazakh, we used a corpus of approximately 300 hours of read speech available through the Open Speech and Language Resources (OpenSLR) web site (https://www.openslr.org/102/) (Khassanov, et al., 2020). In both cases, the additional data significantly improved WER on broadcast speech. Results are shown in Table 3. Note that these improvements are on top of a system that already includes LM expansion and SST.

Table 3. WER Performance Improvement from Additional Transcribed Acoustic Data.

System Description	Pashto	Kazakh
MATERIAL data only	39.2	18.3
+ Additional transcribed acoustic data	36.3	16.7

Given how well the additional out-of-domain data works for the program, we further investigated if it could replace in-domain data transcription in general. Using Kazakh as the test case, we found that replacing the build data with the OpenSLR data actually improved performance on broadcast data; however, performance on CTS data is terrible. Results are shown in Table 4. Effect on WER of Using Out-of-Domain Data for AM Training WER is
the Average over all Data, which is Dominated by NB and TB data.

. Bootstrapping from out-of-domain data to a more difficult domain like CTS is a far more difficult problem than News Broadcasts (NB) or Topical Broadcasts (TB). Note that in the results below, an expanded LM and lexicon is still used.

Table 4. Effect on WER of Using Out-of-Domain Data for AM Training WER is the Average over all Data, which is Dominated by NB and TB data.

System Description	WER	CTS	NB	TB
Build Data only for Acoustic Model	25.9	38.9	22.1	25.5
OpenSLR data only for Acoustic Model	26.9	71.5	19.6	23.6

3.1.5 Additional Untranscribed Acoustic Data

In addition to the acoustic data provided as part of the program, Brno University of Technology (BUT) also collected large amounts of untranscribed audio from YouTube. By the end of the program, we were collecting more than 5,000 hours of data per language. In order to make the best use of the data, we spent a significant amount of time and effort processing the data. This included language identification, first pass decoding, and several methods for selecting the best subset for training. As shown in Table 5, despite all of the effort, the large amount of YouTube data rarely gives larger gains than just using the much smaller set of Eval data.

Table 5. Comparing WERs with YouTube Data and Eval Data for Semi-Supervised AM Training.

Language	Supervised	YouTube SST	Eval SST
Lithuanian	21.3	17.3	18.1
Bulgarian	20.0	17.6	17.7
Pashto	36.5	35.7	34.2
Farsi	36.3	27.6	27.1

Looking more closely at Farsi, the result in Table 6. Comparing WERs on Farsi for SST with Large Amounts of Out-of-Domain Data with Small Amounts of In-Domain Data shows that even just ten hours of data from the Development (Dev) set gives the majority of the gain. In this case the YouTube data set was a 1500 hour set that was sub-selected down to 150 hours based on confidence. It is simpler and cheaper to use untranscribed data from the domain of interest rather than searching for additional untranscribed audio.

Table 6. Comparing WERs on Farsi for SST with Large Amounts of Out-of-Domain Data with Small Amounts of In-Domain Data.

WER is the average over all data, which were dominated by NB and TB data.

System Description	WER	CTS	NB	TB
Supervised Only	39.6	37.2	39.7	40.1
SST on Dev Set	34.6	36.7	32.9	35.0
SST on YouTube	33.0	36.2	30.2	33.8

3.1.6 Lattice Boosting

We utilized lattice boosting to improve CLIR performance. The approach is a post-processing technique applied to the lattices. This means it does not require knowledge of keywords at decode time. Given a set of keywords, we determine likely translations based on our translation table. Any occurrences of the translated queries in our lattices are boosted. This is accomplished by simply adding a constant factor to the log likelihood. While it was difficult to measure impacts on CLIR given the limited number of documents in the ASR analysis and dev sets, it did appear to give a consistent gain. Results in Table 7 show the impact for both Pashto and Farsi.

Table 7. Maximum Query Weighted Value (MQWV) Improvements on the Dev Set from Using Lattice Boosting

Language	No Boosting	With Boosting
Pashto	48.1	52.3
Farsi	52.8	53.9

3.2 MT

In this section, we describe the MT components for the OP2 system, using Georgian as an example whenever specific numbers are quoted.

3.2.1 Data

3.2.1.1 Parallel Data

The parallel data sources for constructing MT models are the parallel data provided by the MATERIAL program and the PanLex lexicon.[1] We also use parallel data from OPUS.[2] However, because some of the OPUS data is of poor quality, we filter the sentences by using our best MT model to translate the foreign side of each sentence and then compare the automatic translation with the provided English translation using TER. We only keep those sentences with TER below some tuned threshold. This method works well even with relatively poor initial MT models. This is because we determine the threshold for TER separately for each language by a combination of manual inspection and running experiments. For the Georgian evaluation, our experiments showed the largest BLEU gain from using a TER threshold of 80, so any reference to OPUS data going forward refers to TER-filtered OPUS data using a threshold of 80.

Additionally, we downloaded a small amount of Georgian-English parallel data found online at the following websites: Goethe-Verlag[3], National Parliamentary Library of Georgia[4], Georgian-English dictionary[5].

3.2.1.2 Monolingual Data

While the amount of parallel data available is very limited for many MATERIAL languages, we can download a lot of monolingual data from the Web and use it in SST. In MATERIAL, we used the BBN WebText collection system (Zhang et al., 2015) originally developed under the IARPA Babel program to automatically retrieve large amounts of monolingual data starting with a small seeding corpus. We use Bing Web Search API[6] to find the web documents to download. Since there is no web crawling involved, a large amount of data can be acquired in a short time. Our tool is capable of downloading 1B words within a day.

For Georgian, we used the foreign side of the parallel data on the Analysis Set, and the MT English output of that as seeding terms when conducting the web retrieval. This is to ensure that the monolingual data we download is close to the test condition. Additionally, we pre-filtered the

---

1 https://panlex.org
2 https://opus.nlpl.eu
3 http://www.goethe-verlag.com/book2/EN/ENKA/ENKA037.HTM
4 http://www.nplg.gov.ge/gwdict
5 http://b.sisauri.tripod.com/pdf/lexicon.PDF
6 https://www.microsoft.com/en-us/bing/apis/bing-web-search-api

seeding text to keep only words written in the relevant script for the language (i.e., Cyrillic script, Mkhedruli script, or Latin script). The data were then used for SST in the forward and backward translations.

The size of each corpus used in our Georgian MT models is listed in Table 8.

Table 8. Data Used for Building Georgian MT Models

Data source	Size (source tokens)
MATERIAL	22K
Panlex	78K
TER-filtered OPUS	23.8M
Goethe-Verlag	8K
National Parliamentary Library of Georgia	227K
Georgian-English dictionary	5K
Monolingual, semi-supervised, back-translated	21M
Monolingual, semi-supervised, forward-translated	55.4M
Total	**100.5M**

3.2.2 NMT Models

Our MT system employs two standard multi-layer Neural MT models, Transformer (Vaswani et al., 2017) and Dynamic Convolutions (Wu et al., 2019), with a 50k subword vocabulary. We decode both models in an ensemble at decoding time. We used the Fairseq toolkit[7] with some modifications for biased MT, described later. Each MT decoding can produce 1-best output as well as lattices with additional translation paths, both of which are used in our CLIR and Summarization components. We describe the different MT techniques in more detail below.

3.2.2.1 Bi-directional Subword NMT

Our subword MT models are bi-directional neural models that can translate between Georgian and English based on a shared subword vocabulary of 50k units. This is implemented on top of the tagged training concept first introduced in (Johnson et al., 2017). We add a copy of the training data with source and target sentences reversed and the appropriate language tag prepended. The bi-directional NMT removes the need for training separate forward and backward translation models, while yielding better BLEU scores than those obtained from separated models. This is because the model can leverage multilingual training in both directions, for all languages in the training. In an effort to further reduce training complexity, we

[7] https://github.com/pytorch/fairseq

design our NMT model to handle both text and speech decoding tasks. This is achieved by augmenting the training data with a copy of the bitext transformed with case and punctuation marks removed to mimic the ASR output.

3.2.2.2 Training NMT with Per-Corpus Weights

Our MT models for previous evaluations have utilized multilingual training to leverage parallel training data from related languages. Although we did not directly use data from other languages during the Georgian evaluation (because we did not have parallel data from languages that we thought were related to Georgian), we used the same architecture to assign different dataset weights based on how much we want them to contribute to the final model.

Each of our data sources (Panlex, OPUS, etc.) has different amounts of training data. Since the data vary widely in size, the larger datasets will dominate the models as well as the learned subword vocabulary. The resulting model may be biased to that particular data type in the training. As mentioned above, the different data sets also have data of differing quality. To address these problems, we oversample sentence pairs from high-quality but low-volume datasets so that they are well represented in the training. The resulting model is more balanced and performs better across a variety of test sets. We start with initial weights based on our intuition about the relative quality of each corpus (e.g., we expect Panlex to be higher quality than OPUS). We then tune the oversampling weights by trial and error in multiple experiments to optimize performance on blind test sets.

Table 9 shows the tuned oversampling weights used for base and SST MT models during the evaluation. The weights add up to 1, so a weight of 0.1 for a corpus means that this corpus made up ten percent of the final training data after oversampling. An empty box means that the model did not use that particular dataset during training.

Table 9. Per Dataset Weights Used in Training Georgian MT Models.

*The semi-supervised and mono/cross IR methods of acquiring data are described further below in Section 3.2.3***.**

Data source	Base model	SST model
MATERIAL	0.10	0.01
Panlex	0.10	0.10
OPUS	0.60	0.27
Goethe-Verlag/NPLG/Geo-Eng dictionary	-	0.02
Monolingual, semi-supervised, back-translated	-	0.20
Monolingual, semi-supervised, forward-translated	-	0.20
Mono/cross IR (no SST)	0.20	0.20
Mono/cross IR (including SST)	-	-

3.2.2.3 High Recall Subword/Word Hybrid NMT for Summarization

In our Foreign Language Automated Information Retrieval (FLAIR) system, the primary use of MT is for Summarization. For the Summarization task, the percentage of the query words in the MT output plays a much more important role than BLEU score when it comes to the AQWV metric. If a query word is missing from the MT output, no amount of post-processing can recover it after MT decoding. By the same token, the Summarization system will not select an MT summary snippet if it does not contain the query word. It is therefore desirable to design an MT system that can give high query word coverage.

To that end, we construct a hybrid NMT model where the source languages are modeled at the subword level but word level tokens are used for the target language. The subword unit on the source side enables us to take advantage of multilingual training, using the same subword embedding layer for all source languages. It also allows the MT system to translate words that were not present in the parallel training. The whole-word output makes it easy to produce word lattices in the FST format. The word lattice output allows us to bias the query words in the output in different ways, which will be described later. To ensure high query word coverage in the NMT output, we used a large target vocabulary of 70k English words. We picked the vocabulary size based on measuring the coverage of the Query set 1 (Q1) query words with different vocabulary sizes.

3.2.3 SST

To further improve MT, we employ semi-supervised training to leverage the large amount of monolingual WebText we downloaded in both Georgian and English. Once an initial bidirectional MT model is built, we use it to forward and back translate the Georgian/English monolingual data, thereby producing semi-supervised training data. We prepend different tags to SST training data to signal to the model that those data should be treated differently from the original parallel data. We repeat the steps of training the model and generating new SST parallel data twice to produce the final Full SST MT model. The partial SST MT model does not include the full SST output in training, but the mono/cross-lingual IR is run over the SST data as well as the supervised data, so a portion of the data ends up being used in training.

3.2.3.1 Duplicating Training Sentences Containing Query Words

In order to ensure that the MT model would translate the query words well, we run monolingual and cross-lingual IR on the training data to find sentences that contain the query words. We then replicate these query-relevant sentences in the training, so that the model gives slightly higher weight to the query words. Although this does not significantly affect the BLEU score, we see improvements on the CLIR task using the new trained model. During the Georgian evaluation, copies of 18.5M additional words were added when we searched through just the Program-supplied BUILD pack, Panlex, and OPUS data sets. When we included the semi-supervised data in the search, we added 57.7M words to the training.

3.2.4 Biased MT for Summarization

We use a few techniques to bias the MT output to increase the chance of query word appearing in the output: Dynamic Boosting, FST lattice biasing, and Constrained Decoding.

3.2.4.1 Dynamic Boosting

Dynamic Boosting runs after CLIR on those sentences that CLIR believes have source words that could translate into the query words. The "boost" is a factor that is multiplied by the output layer for a particular word, making it more likely to appear in the 1-best output. Note that the query often consists of multiple words. The algorithm decodes each sentence multiple times. Each time, it adjusts a Boost factor for each of the query words in a logarithmic binary search to determine the lowest boost value needed for each query word to appear in the 1-best MT output. The result is that the final output usually (about 95% of the time) contains one token of each of the required query words, appearing in a place that makes sense within the whole sentence.

3.2.4.2 FST Lattice Biasing

We modified the decoder in the fairseq toolkit to generate word lattices, which are then saved in the standard FST format. We then compose the lattice with a query-specific FST to increase the likelihood that the desired query words appear in the output. Finally, we find the 1-best path through the composed FST lattice. The resulting output contains all of the query words about 90% of the time using this method.

3.2.4.3 Constrained Decoding

We also used a lexically constrained MT decoding approach implemented in Sockeye toolkit[8]. This idea is similar to dynamic boosting but restricts the iterative process to the beam search. The output contains all of the query words about 99% of the time. However, this method can cause too many query words to appear in the translation, leading to many repeated query words. Therefore, we only use this output in Summarization if the other methods fail to output the query words.

In addition to the output from the biased MT above, we also generate a low-weight boosted MT with a low degree of coverage, as well as the normal subword 1-best MT output as additional input to Summarization. The many MT outputs, each operating at a different query word

[8] https://github.com/awslabs/sockeye

coverage level, give the Summarization component the flexibility to choose the best snippet to be presented to the user based on CLIR scores.

3.3 CLIR

3.3.1 System Description

The CLIR component of the E2E system is primarily based on a probabilistic framework originally introduced by BBN (Xu et al., 2000): both documents and queries are represented as "bags of words" and relevance of a document with respect to a query is accumulated linearly over the document – no handling of the special format of the queries and no extraction of rich structure from documents is done. The equation for computing the retrieval score is as follows:

$$P(doc\ is\ rel \mid Q) = \prod_{q \in Q} \left(\alpha \sum_{f \in doc} \frac{P(q|f)}{|doc|} + (1-\alpha) P_{LM}(q) \right)$$

where Q represents the set of query words, $P(q|f)$ is the probability of translating foreign word f into query word q, P_{LM} is a general English language model, and α is an interpolation constant.

This model was subsequently augmented with several innovations developed over the four years of the MATERIAL program, as described below.

3.3.1.1 Bag-of-Words Probability-of-Occurrence Retrieval Model

This is a more advanced version of the bag-of-words framework. It computes the probability that a document contains at least one occurrence of the query (Zbib et al., 2019), and it is more in line with the criterion of relevance in MATERIAL (a document is considered relevant if it mentions the query at least once). The formula for computing the probability that a query appears at least once in a document is given by:

$$P(query\ appears\ at\ least\ once) = \prod_{q \in Q} \left(1 - \prod_{f \in doc} (1 - P(q \mid f)) \right)$$

3.3.1.2 Hierarchical Retrieval Model

To better handle the complex nature of the queries, we developed a *hierarchical* retrieval model, which parses the queries with a context-free grammar (provided by IARPA) and then uses the parse tree as a bottom-up computation tree (akin to the *abstract syntax tree* used in computer language compilers) (Zhang et al., 2020). This hierarchical retrieval mechanism allows us to perform more accurate retrieval for two types of queries: phrase queries and conceptual queries. More details about these two types appear below.

3.3.1.3 Proximity Matching for Phrase Queries

Phrase queries, which consist of multiple words, are detected more effectively when the proximity of translations of these words in the document is taken into account. For this to work, detailed information about the exact location of the words in the document is saved during indexing. At retrieval time, the probability of detecting the phrase query becomes a weighted average of three products of probabilities: the product of the probabilities of occurrence of the phrase words within a document, within the same sentence, or within a window of each other in the same sentence. In mathematical terms,

$$P(x\,y) = \alpha\, P_{occ}^{doc}(x) \cdot P_{occ}^{doc}(y) + \beta\, P_{occ}^{sent}(x) \cdot P_{occ}^{sent}(y) + \gamma\, P_{occ}^{win}(x) \cdot P_{occ}^{win}(y)$$

where x, y are the words of a phrase, P_{occ}^{doc} is the probability of occurrence in the entire document, P_{occ}^{sent} is the probability of occurrence in a sentence, and P_{occ}^{win} is the probability of occurrence in a specific location in the document. (After computing the above products, one more probability-of-occurrence computation is done over the entire document.)

3.3.1.4 Whole-Phrase Matching for Phrase Queries

Improved retrieval of phrase queries can also be obtained using whole-phrase retrieval, where the probability of translation of an entire phrase is estimated using a standard SMT toolkit (Moses) (Zhang et al., 2020). At retrieval time, the probability of detecting the phrase query is the weighted average of the proximity-based probability mentioned above and the whole-phrase retrieval. In mathematical terms,

$$P(x\,y) = \theta_1\, P_{occ}(x) \cdot P_{occ}(y) + \theta_2\, P_{occ}(x\_y)$$

where $P_{occ}(x\_y)$ represents the probability of occurrence of the multi-word string x_y based on phrase translation probability.

3.3.1.5 Query Expansion for Conceptual Queries

Conceptual queries are queries that express a "topic", and the relevance requirement of MATERIAL is that documents should contain that topic. Query expansion, where the system searches for other words that are related to the query of interest, is an effective way of improving the performance of retrieval. We have experimented with two approaches for finding such related words: (i) using nearest neighbors in an embedding space (Zhang et al., 2020); (ii) using co-locations computed with a large English corpus (Wikipedia), scored using pointwise mutual information. Although query expansion can be used with the bag-of-words model, it is more accurately done under the hierarchical model because the latter can better handle the retrieval of expanded phrases, as mentioned above.

3.3.1.6 NNLTM

We designed and implemented the NNLTM which allows us to come up with *contextualized* lexical translations of foreign words into query-language (English) words (Zbib et al., 2019). This model very effectively complements the *context-independent* translation probabilities obtained through standard statistical approaches (Expectation-Maximization, using GIZA), resulting in large gains in performance. During indexing, we compute the probability of query word (at a specific location in the document) as

$$P(q) = \alpha \sum_f P_{GIZA}(q|f) + \beta\, P_{NNLTM}(q)$$

3.3.1.7 Indexing NMT 1-Best and Lattices

In addition to indexing foreign documents using lexical translation tables or context-dependent neural lexical translations, we use a unified framework to index the output of a neural machine translation system. The NMT output is in the form of 1-best (a sequence of English words per foreign sentence) or the form of a lattice (a rich hypothesis space that encodes multiple sequences of English words per foreign sentence). During indexing, we compute

$$P(q) = \alpha \sum_f P_{GIZA}(q|f) + \beta\, P_{NNLTM}(q) + \gamma\, \mathbf{1}(q\ in\ MT) + \delta\, P_{lat}(q)$$

3.3.1.8 Data Augmentation for Estimating Improved Lexical Translation

We spent a considerable amount of effort in augmenting the supervised parallel training data with automatically generated data. The augmented data were used to train more accurate NMT models and lexical translation models (both context-independent and context-dependent). The techniques we experimented with were: (i) forward and backward translation of monolingual data; (ii) a transliteration mechanism for handling OOV query terms; (iii) pivoting through a third language for both generating automatically translated data and for summing-out the probabilities (see paraphrase work of Callison-Burch et al., 2006) and (iv) emphasizing parallel sentences that contain the query words by performing IR on the training data.

3.3.1.9 CLIR Over Speech Documents

The standard approach for performing CLIR over speech documents is to apply an ASR system on the speech, convert the speech into text (1-best output), and then use the CLIR system in the usual way. However, as we demonstrate in (Zbib et al., 2019), using a richer hypothesis space (lattice or confusion network) performs much better, as it offers correct alternatives that are missing from the 1-best. The retrieval score is obtained by weighting the translation probability by the source-word confidence provided by the ASR system (Zbib et al., 2019).

3.3.1.10 Score Normalization and Thresholding

As mentioned elsewhere, one of the requirements in MATERIAL is to decide on what documents from the final ranked list to give to the analyst. This is done with a thresholding process: all documents with retrieval score above the threshold are selected for downstream processing. The thresholding is done over the entire set of queries; this means that the retrieval scores of all documents over all queries have to be commensurate with each other. To do that, we use the *score normalization* process mentioned in (Karakos et al., 2013) and (Karakos et al., 2020), where the retrieval scores are transformed according to a query-specific thresholding (QST) scheme. QST is based on decision theory and can be shown to be optimal under some simplifying assumptions. This baseline scheme has worked very well for Text CLIR. For Speech CLIR, as we mention later in the report, there has been a significant performance gap from the optimal thresholding in the last year of the program; for that reason, we introduced algorithmic improvements such as Gaussian mixture modeling (GMM) and a reduced relative acceptance rate.

3.3.1.11 Expectation-Maximization for Translation Table Interpolation

During the development phase of the system for each one of the MATERIAL languages, we collect parallel data from various sources on the web or we generate parallel data in an automatic way. In the first two phases of the program, we used to put all the data together (with repetition of some of the data for emphasis) and estimate a single translation table. This was a slow and cumbersome process, as the amount of repetition was tuned based on the final CLIR score. To avoid this, we came up with a process that runs the Expectation-Maximization (EM) algorithm, which estimates interpolation weights with which data-dependent translation tables are combined. The objective of EM is to maximize the likelihood of English words given the foreign words in a held-out parallel corpus:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{argmax} \prod_{j=1..M} \left[\sum_{i=1}^{K} w_i \cdot p_i(e_j \mid f_j) \right]$$

(The above *argmax* cannot be computed directly in an efficient way, and that's why we resort to the EM algorithm.)

3.3.1.12 Efficient Pruning during Search

The search for conceptual queries is computationally expensive, because (to obtain the best possible performance) we perform query expansion. There are queries for which the expansion introduces *two orders of magnitude* more foreign-translation alternatives than without expansion; naturally, this leads to a significant throughput reduction. To deal with this problem, we implemented two pruning mechanisms, which improved throughput significantly without causing any system performance degradation: (i) no expansion for named entities, since named entities are usually required to exist in the document verbatim, and (ii) for multi-component queries, down-selection of candidate documents based on the non-conceptual query component (since all query components have to have evidence in a document for the document to be considered relevant, performing a fast retrieval using the non-conceptual query component narrows down the search tremendously).

3.3.2 Experimental Results

In this section, we present results based on a number of controlled experiments on Kazakh Text CLIR, which we ran to answer the following important questions:

a) What was the impact of the various CLIR innovations on AQWV performance over the four years of the program?

b) Given that NNLTM is the single most effective way to improve the performance of the CLIR system, how is the gain broken down as a function of the query type?

To answer (a), we performed a series of experiments with increasingly complex versions of the system. Table 10 shows the Dev MQWV and Eval AQWV (blind testing) as a function of the various CLIR innovations. The first three rows correspond to the bag-of-words retrieval model, while the rest correspond to the hierarchical retrieval model. (The row *other** refers to innovations that are more data-based: using transliterations for OOV query matching, pivoting through Russian, and searching the training data for query words). As can be seen, the single most significant gain is obtained with NNLTM, even after using a number of innovations to improve specific query types. Over the four years of the program, the overall gain has been more than 20% absolute, which we believe is very significant, given that we started with a state-of-the-art system.

Table 10. Dev MQWV and Eval AQWV Results on Kazakh Text with Variants of the CLIR System.

Earlier, simpler versions of the system appear towards the top of the table, while more complex versions appear at the bottom. The number in parentheses is the absolute gain compared to the row above

System	Dev MQWV	Eval AQWV
probabilistic model	60.0	60.4
prob. of occurrence model	63.8 (+3.8)	63.8 (+3.4)
+ query expansion	66.2 (+2.8)	65.0 (+1.2)
+ proximity matching	67.5 (+1.3)	66.4 (+1.4)
+ whole phrase match	69.8 (+2.3)	67.1 (+0.7)
+ NNLTM	76.5 (+6.7)	75.0 (+7.9)
+ other*	80.3 (+3.8)	81.1 (+6.1)
+ NMT lattices	81.7 (+1.4)	82.5 (+1.4)
Total gain	**21.7**	**22.1**

Given the significance of NNLTM, we next wanted to better understand how various query subsets are improved by it. Table 11 shows the per-subset improvements over GIZA alone. As can be seen, the gain is much larger for single-word queries (queries with single-word components) than for phrase queries or Plus (conceptual) queries. One of the reasons could be that the IR for these categories of queries (phrases and Plus) are already improved by other methods such as whole-phrase retrieval and query expansion; so NNLTM has less room to improve. Another (independent) reason, however, could be that NNLTM output probabilities are not appropriate for the multiplicative nature of phrase queries (that is, the product of the probabilities of all N phrase words appearing in the document). We intend to study this question further, in the context of a different project.

Table 11. Dev MQWV Results on Kazakh Text over Various Query Subsets.
The number in parentheses is the absolute gain/loss obtained from using a combination of GIZA and NNLTM, over using GIZA alone

Query subset	GIZA only	GIZA+NNLTM
one word	60.1	72.8 (+12.7)
two words	71.6	77.3 (+5.7)
three words	81.2	82.5 (+1.3)
only single-word components	65.0	75.2 (+10.2)
at least one phrase component	79.6	79.4 (-0.2)
Plus component	84.3	84.5 (+0.2)

3.3.3 Interplay between Precision and Recall as a Function of Beta

In our software delivery to MIT Lincoln Laboratory, we included the plot shown in Figure 1, which shows the relationship between precision and recall of our delivered system as a function of beta, the penalty for false alarms. Clearly, as beta increases, precision increases (fewer documents are accepted) and recall decreases (there are more misses). Although the delivered system comes with a default value of beta=600, it gives the user the option to change it according to his/her precision/recall requirements.

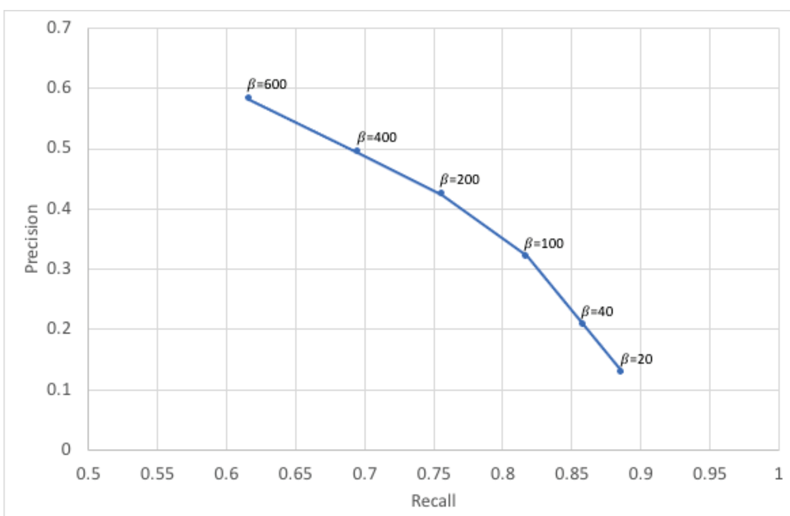

Figure 1. Relationship between Precision and Recall as a Function of Beta.

3.3.4 Thresholding for Speech CLIR

In the Kazakh and Georgian evaluations, the speech evaluation data had a very different "richness" (relevant document density per query) than that of the Dev data, or that of previous languages. So, using a decision threshold based on a pre-determined number of documents per query is not a good strategy. Furthermore, using the well-known QST method (Karakos et al., 2020) is suboptimal. We came up with the following solutions:

- Reducing the number of accepted documents by a tuned fraction (20%)
- Using a Gaussian mixture model to compute the threshold based on decision theory. The Gaussian mixture components model the irrelevant and relevant documents, respectively, as shown in Figure 2.

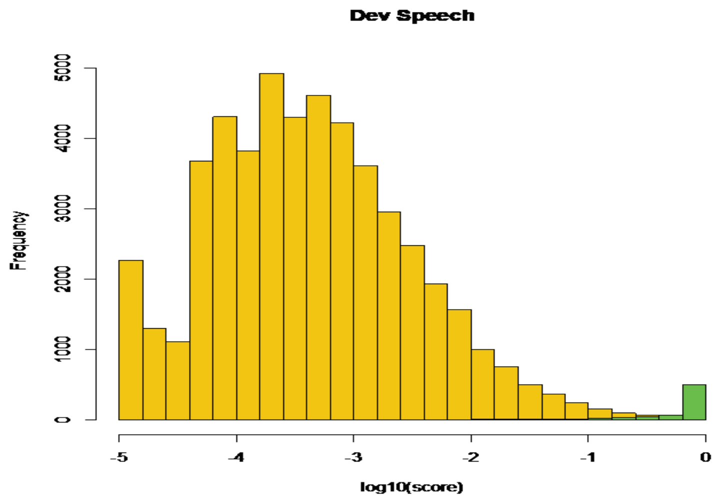

Figure 2. Distributions of Irrelevant (yellow) and Relevant (green) Document Retrieval Log-Scores.

Table 12 shows the results we obtained on Kazakh and Georgian speech CLIR evaluation data (post-eval) using the aforementioned techniques.

Table 12. AQWV Results on Kazakh and Georgian Eval Speech with Various Thresholding Schemes

Technique	Kazakh AQWV	Georgian AQWV
Regular QST	75.0	69.1
Modified QST (20% less)	76.1	73.1
GMM	77.0	74.1
Oracle	77.4	77.8

3.4 Summarization

3.4.1 General Summarization Mechanism

Our basic summarization system (the one that we delivered) uses sentence-level query word evidence from CLIR and looks for the respective query word in the English MT sentence (from one or more MTs). It creates fixed length snippets around the query words, and uses a snippet selection algorithm using the submodular objective that optimizes both coverage and diversity in the summary. In the submodular objective, we incorporate high bias/weights for snippets containing query words in order to incentivize selection of such snippets in the final summary. Finally, the top two to five such snippets are used to form the final English summary with the query words in the summary highlighted. A System Confidence score is also included in the summary, and the instructions to Turkers are customized based on the characteristics of the query (simple lookup or topical lookup) and the summary (query word present or absent).

3.4.2 Summarization for Variable Beta (using Query Biasing)

From Farsi Eval summaries, we observed that the Amazon Mechanical Turkers' decision of accept or reject for a summary had a high correlation with the presence or absence of the query word in the text of the summary and did not depend very much on whether the corresponding document was actually relevant. We measured the probability of rejection of a summary given its degree of query word coverage and actual relevance (we call this pReject). The degree of query word coverage of a summary is a variable with three possible values:

- All (i.e., all query words are present in the summary),
- Some (i.e., some but not all query words are present in the summary), and
- None (i.e., no query words are present in the summary).

Table 13 shows the pReject values (in percentage) for both truly relevant documents (TD) (TD-reject) and false accepts (FA) (FA-reject) for Text documents and lexical queries. As can be seen, when *all* query words are present in truly relevant documents (TD), nearly 96% of the summaries are accepted (TD-reject is 4%). Conversely, when *no* query words are present in truly not-relevant documents (FAs), nearly 99% of the summaries are rejected. These extremely skewed values of pReject for TDs and FAs give us the confidence that actual document relevance and query-coverage provide effective features to model the probability of Turker-rejection.

Table 13. TD and FA Reject Rates for Various Query Coverage Conditions.

Query Coverage Condition	% TD-reject	% FA-reject
All query words present	**4**	**16**
Some but not all query words present	**42**	**86**
No query words present	**86**	**99**

Since the actual relevance of a document is not known for a new (unseen) Eval set, we used the CLIR score of a document as a proxy for actual relevance. To produce different degrees of query coverage, we used a variety of MT systems with varying levels of query-biasing. Below, we show the five methods with the resulting *average* query coverage in parentheses.

- Sockeye MT: Very high degree of coverage (99%)
- Dynamic Boosting MT: High degree of coverage (95%)
- Lattice Biasing MT: Moderate degree of coverage (89%)
- Subword MT: Unbiased best-BLEU MT (76%)
- Negative Boosted MT: Very low degree of coverage (<1%)

We produce different biased MT versions for each document. Then, using the combination of CLIR score, degree of query coverage, and pReject, we predict which MT version (or degree of query coverage) is likely to yield the correct Turker decision (accept/reject) for each document.

Table 14 shows the E2E AQWV scores for the official evaluations on OP-2 languages – Farsi, Kazakh and Georgian – (CLIR AQWV scores are also shown for comparison). Note that for Kazakh and Georgian, the E2E AQWV was calculated with Beta=600 (whereas, for all other conditions, Beta was 40).

Table 14. E2E Results for OP2 Languages.

Language	Genre	CLIR AQWV	E2E AQWV
Farsi	Text	79.9	75.6
	Audio	69.5	62.5
Kazakh	Text	82.5	47.5
	Audio	69.7	27.2
Georgian	Text	80.6	48.9
	Audio	73.1	31.1

3.5 Tech Transition

Raytheon BBN offers a product called the M3S, which enables access to video, radio, web, and social media in 22 different strategic languages in real time. Sources are user-selectable and can scale to years of video and radio, and millions of web pages and social media posts. It provides powerful visualizations including charts, geotagging, and social diagrams to explore data. It is used operationally at United States Special Operations Command (USSOCOM), United States Central Command (USCENTCOM) and other agencies domestically and internationally. All speech is automatically transcribed in the source language and all text is then automatically translated to English. The data is made accessible through simple CLIR, so that users can look for foreign documents of interest using English queries. Retrieved documents can be displayed showing the foreign source and the English translation together.

The system was improved by adopting two MATERIAL technologies: NMT and improved CLIR. The NMT resulted in significantly improved MT quality and it now translates all foreign words to English. While the original system retrieved documents by simply matching the English queries against the English translations, it now searches directly in the source language for various translations of the English query words, which finds more relevant documents. The retrieved documents are shown in both languages, with related words in both the source language and English highlighted.

4.0 PROGRAM RESULTS, FINDINGS, AND TECHNICAL INSIGHTS

4.1 Successes, Partial Successes, and Failures

4.1.1 ASR

The biggest success in ASR was the use of automatically collected web data for language model and lexicon expansion. It was not just a matter of reducing the OOV rate. The text also improved the overall language model. WER reduction was as high as 20 points absolute. Collecting transcribed in-language data was also a consistent success. Even in the case of Kazakh where the data were simply read speech, it significantly improved performance. When using untranscribed data, it was critical for the data to be in-domain. Even ten hours of in-domain data were enough to produce large improvements in WER. Lattice boosting was our best approach to improving CLIR performance directly without improving WER. All other techniques improved CLIR performance by reducing the WER of the system.

We also explored several avenues that ultimately did not impact performance. Using out-of-domain, untranscribed audio (e.g., YouTube) was of limited help. For Bulgarian and Lithuanian, it provided a modest improvement on our best system, but was no help on other languages. Obtaining any improvement required the use of significant time and computational resources. We explored a variety of approaches to better filter and select the data, but that direction did not prove fruitful. Our standard approach to acoustic model pretraining uses a large amount of multilingual conversational telephone speech. Since the focus in MATERIAL was on the broadcast domain, we also explored pretraining the model on broadcast data. Despite being a closer match to the correct domain, this step provided no benefit.

4.1.2 MT

Successes: The big success in MT has been multilingual training for Neural MT. In the Base period, we showed that Neural MT can be very data hungry and did not work as well as Statistical MT in a low-resource setting. This changed in OP1 when we deployed multilingual training where one can easily leverage training data from related languages through a shared subword vocabulary. This not only allowed us to have a full Neural system that performs better than any SMT we could built, but also paved the way for other innovations, such as per-dataset weighting scheme, tagged bi-directional training, and semi-supervised training with forward/back translation. This direction proved to be very successful and we were able to completely switch to NMT in OP1 with improved performance across all languages.

Another success was the BBN-designed hybrid subword/word NMT model. This enabled us to use different query-informed decoding schemes to effectively boost query word coverage in the MT output by 30 points absolute: 95% of query words can be covered in our summarization system compared to the usual 65-70% range without biasing.

Partial Successes: We had partial success when it came to model fine-tuning. Most of the time, the gain was smaller than we expected. We also tried fine-tuning the 25-language multilingual Bidirectional and Auto-Regressive Transformers (mBART) model[9] from Facebook with limited success: the gain we saw in low-resource conditions disappeared quickly once the amount of parallel data increased. The Farsi Twitter out-of-domain adaptation exercise confirmed that model fine-tuning is something we should look into improving.

[9] https://github.com/pytorch/fairseq/tree/main/examples/mbart

Failures: During the OP1 period, we tried normalizing all MT training data using the Universal Romanizer[10] on Lithuanian and Bulgarian, with the expectation that Romanization would help normalize morphological variations that existed in the training. Unfortunately, we observed no gain. We then tried the idea on Kazakh; again there was no gain. We suspect that the sentencepiece subword tokenizer from the Google library already clusters related morphological text segments into the same subword unit. Therefore, romanization offered no gain.

4.1.3 CLIR

Successes: One big success is the AQWV improvement of over 20% absolute between the initial system of the Base Period and the final system of OP2. A significant contributor to this success was the development and effective deployment of the NNLTM, which was the single most effective innovation developed by our team, resulting in gains on the order of 8% absolute on a blind test set. Other successful algorithmic innovations include the probability of occurrence retrieval model, the hierarchical retrieval model, query expansion, whole phrase matching and proximity-based phrase matching, NMT lattice generation and indexing, and deciding what documents to give to the analyst, based on a principled thresholding scheme. In terms of data, successful innovations included pivoting through a third language, handling of OOV queries through transliteration, and data weighting through replication of parallel data sentences that contain query words.

Partial Successes: One aspect of the CLIR system that was not entirely successful was the thresholding for Speech documents (deciding what Speech documents to give to the analyst). Although the aforementioned principled thresholding scheme worked very well on Text documents, it did not work well on Speech documents, especially in the last two languages of the program (Kazakh and Georgian) where the proportion of queries with relevant documents between the Development and the blind Test data was very different. We developed techniques to deal with this mismatch; the gap in performance from the optimal (oracle) thresholding was reduced but it remained quite high for Georgian Eval Speech.

Failures: Our efforts to create enough, or good quality, parallel data (for building better translation models) by collecting data from the web were not successful. We spent a significant amount of time experimenting with the openly available tool *Bitextor*, but CLIR performance did not improve significantly. Furthermore, efforts to use neural architectures based on Bidirectional Encoder Representations from Transformers (BERT) for CLIR did not give significant gains by themselves or in combination with the statistical (+NNLTM) approach, and definitely not as large gains as those reported in the literature for monolingual IR tasks. This is something that has perplexed us (as well as others in the community): despite the tremendous success of BERT-based IR approaches in a monolingual setting, BERT-based CLIR (in a cross-lingual setting) has lagged behind statistical methods.

4.1.4 Summarization

We obtained our best results for Summarization by developing an AQWV optimization algorithm that chooses, from among multiple versions of query-biased MT outputs, the version that increases the likelihood of correct Turker decisions (accept for TD or reject for FA). This algorithm can be tuned for any given value of Beta for the E2E human judgments. We achieved this success primarily in the last phase of the program.

---

[10] https://github.com/isi-nlp/uroman

In the base phase, we approached the problem as the classical query-informed document summarization problem. However, it was soon evident that that was not the right approach for the task at hand. In OP1, we made several improvements to the system—shorter summaries that better suited the Amazon Mechanical Turk (AMT) paradigm, using inputs from multiple MT and CLIR systems, and using footnotes to give hints about alternative translations of the query words. While our performance did improve with all these measures, the overall gain was not very significant.

Finally, for OP2, we started looking at the problem as an optimization problem, where data from previous AMT evaluations could be used for learning. It was this approach that not only gave our system significant boost in performance, but also placed us ahead of the other teams.

4.1.5 Tech Transition

As described in Section 3, we improved the M3S system by transitioning NMT and a version of CLIR developed under MATERIAL. The quality of the translations displayed are significantly improved and no longer contain untranslated foreign words. The recall of the retrieval is significantly improved by virtue of using translations of the query, and the highlighting of both the source and English side of the display was improved to include words related to the query.

4.1.6 Delivery

We successfully delivered all of the components that were required. This includes an ASR system for Tagalog, an NMT system that can translate any of the nine MATERIAL languages and can perform fine-tuning given additional parallel data, and a full E2E Farsi system that can process data like an Evaluation set, performing MT, CLIR, and Summarization. All delivered Dockers were thoroughly tested on Amazon Web Services (AWS).

4.2 Hardware, Software, External Data, and Compute Requirements for OP2

This section lists the hardware, software, external data, and compute requirements to train and evaluate OP2 models.

4.2.1 Hardware Requirements

BBN's system runs over a cluster of hundreds of off-the-shelf x86-64 computer servers with varying amount of central processing units (CPU) cores, memory, and Nvidia Corporation Graphics Processing Units (GPU) (K40-V100) for numerically intensive computing. In total, we have access to over 2,000 CPU cores and about 100 GPUs at any given time.

4.2.2 Software requirements

Our base software stack consists of:

- CentOS 7 Linux (https://www.centos.org/)
- Grid Engine (https://arc.liv.ac.uk/trac/SGE)
- Singularity container (https://sylabs.io/singularity/)
- Nvidia CUDA 10.0 GPU runtime

Most of our software runs inside a Singularity container on top of the base environment. With additional external software from:

- Kaldi ASR toolkit (https://www.kaldi-asr.org/)

- Python PyTorch module (https://pytorch.org/)
- Python NLTK module (https://www.nltk.org/)
- Python Fairseq module (https://github.com/pytorch/fairseq)
- Python scikit-learn module (https://scikit-learn.org)
- Python Sockeye module (https://aws.amazon.com/blogs/machine-learning/train-neural-machine-translation-models-with-sockeye/)
- Python tensor2tensor module (https://github.com/tensorflow/tensor2tensor)
- Python sentencepiece module (https://github.com/google/sentencepiece)
- Python Scrapy module (https://scrapy.org/)
- Python Beautiful Soup module (https://www.crummy.com/software/BeautifulSoup/bs4)

Table 15 and

Table 16 show the approximate total GPU, CPU, and elapsed times for each of the major systems for training models and for decoding and processing the evaluation data, respectively. Each of the items was aggregated over many jobs; some ran serially, and some ran on a large number of CPUs. In several cases, different detailed processes are grouped by the major categories shown.

Table 15. Approximate Total GPU, CPU, and Elapsed Training Times for each of the Major System Components Used in Training the OP2 System.

Training Processes	GPU hours	CPU hours	Elapsed time (days)
ASR	70	2,400	2
MT	1,000		7
GIZA (CLIR)		58	1.2
NNLTM (CLIR)	39	70	2
CLIR indexing and tuning		630	1
Total Times (approximate)	1,100	3,100	13

Table 16. Approximate Total GPU, CPU, and Elapsed Decoding Times for each of the Major System Components Used in the OP2 Evaluation.

Decoding/Evaluation Processes	GPU hours	CPU hours	Elapsed time (days)
ASR		100	0.25
MT	73		0.8
GIZA (inside CLIR)			

NNLTM (for CLIR)	8		1 hour
CLIR (search, GIZA, comb., normalization, etc.)		10,800	0.5
Summarization:			
Lattice biasing		1,874	0.8
Create summaries		85	0.1
Total Times (approximate)	81	13,000	2

Table 17 shows the amount of monolingual data (number of words) we downloaded using the BBN WebText Collection System for the OP2 languages.

Table 17. Size of Monolingual Data Acquired for OP2 Languages, either from the Open Web or Constrained to News Websites.

Language		Open Web	Web News
3S	Farsi	765 MW	625 MW
	English	N/A	99 MW
3C	Kazakh	472 MW	389 MW
	English	N/A	40 MW
3B	Georgian	189 MW	320 MW
	English	N/A	95 MW

4.3 Experience with Docker Deliverables

Because of the mismatch between BBN's runtime environment and that of the delivery target, we spent a lot of time to simplify our Farsi E2E system so that it runs on a single Docker machine. We also ran into some issues when testing the Docker image on AWS. The target AWS instance type p3.8xlarge, a machine with 4 Nvidia V100 GPUs and 32-CPU cores, is so popular and oversubscribed that sometimes we had to wait for hours before one was available in our AWS zone.

4.4 Possible Program Changes

4.4.1 Program Metric

We believe the program would have been better served using one of the mainstream ranked retrieval metrics, such as mean average precision (MAP), which evaluates a ranked list of documents for each query separately. We believe this metric better reflects real usage. MAP has the advantage that it avoids the added complications of score normalization and setting thresholds that are needed with AQWV. Also, there is a strong possibility that, with MAP, using human post-filtering to remove false alarms might actually yield a significant improvement in performance. In addition, there is a much higher likelihood of having our papers accepted for publication when using MAP instead of AQWV.

4.4.2 Constrained Data Condition in Evaluation

In a program that simultaneously encouraged the finding of more data to train on, as well as enhancing the component technologies, it would have been beneficial to isolate the effects of more data from advances in the component technologies. That could have been accomplished by establishing two evaluation conditions: (1) an *unconstrained condition*, where teams could use whatever data they were able to find, and (2) a *constrained condition*, where the amount of training data was fixed across teams. This would have allowed the program to distinguish the benefits achieved by finding more data versus those resulting from improved algorithms.

4.4.3 Have Humans Judge all Summaries

Given the manageable number of summaries generated by evaluation systems, we think it would have been preferable to have all summaries judged once rather than judging one third of the summaries with three judges each. Judging all summaries would give more statistically valid measures of performance and would avoid the problem in extrapolating performance from a subset of the summaries.

5.0 TRANSITIONS

Below is a list of artifacts and transitionable components that BBN can offer for use by the Government, organized by topic area.

5.1 Data Collection and Processing

1. BBN Webtext Collection System

This is a turn-key system for downloading large amounts of mono-lingual data from the Web using the Bing Web Search API from Microsoft Azure. It's available as a Docker container. The software will typically download about 1B words of web text in one day, using a small corpus of the language as a seed.

2. Program to Filter Collected Monolingual Text to obtain Higher Quality Monolingual Data

This program, which is part of (1), uses a simple language identification module that selects data based on a word list from that language. One use of the program is to filter the data collected by the Webtext Collection System to obtain data of higher quality for later use.

5.2 Data

3. Filtered Text for MATERIAL Languages

We can provide the downloaded data for the languages we have worked on.

   a. Filtered monolingual text used for ASR vocabulary and Language Model
   b. Panlex dictionary
   c. Additional parallel text
   d. Monolingual news in the Foreign Language (for OP2 languages)
   e. English news about that country (for OP2 languages)

5.3 Probabilistic CLIR

4. GIZA Translation Tables for Pashto, Farsi, Kazakh, Georgian

Tables estimated from supervised and unsupervised parallel data. For each foreign word, the tables contain the probabilities of English words. These tables are used by CLIR to compute the probability that a document is relevant to an English query.

5. Phrase Translation Tables for Phrase-Based CLIR for Pashto, Farsi, Kazakh, Georgian

Similar probabilities to (4), but for foreign phrases and English (query) phrases.

5.4 ASR

6. ASR Models (compatible with Kaldi and PySpeech) for the various languages; includes AM, vocabulary, and LM for each language

5.5 NMT

7. NMT models (multilingual, bidirectional) estimated from supervised and unsupervised parallel data

    a. The NMT models include training from related languages, which we find improves performance when the amount of training for the source language is limited. For example, for Pashto, we included training from Arabic and Farsi.

    b. Bidirectional models are used to create synthetic quasi-parallel data from monolingual English news from the target country and monolingual source text, which are then used for training improved MT models.

8. NMT fine-tuning component

    Fine-tunes pre-trained NMT models on a particular data set for better performance. Models trained broadly on multi-lingual parallel text can be made more appropriate to a particular language and domain by fine-tuning on a smaller amount of parallel text that is closer to the domain of interest.

5.6 Optimization

9. Program to determine optimal combination of judgment scores and CLIR scores

We have already delivered the code that combines the AMT human judgment scores with CLIR scores to get superior AQWV performance. It is tuned on a development set. This is specific to MATERIAL. It will run with any type of CLIR scores, as long as higher scores mean better scores.

6.0 CONCLUSIONS AND RECOMMENDATIONS

6.1 ASR

The best systems used in the MATERIAL program were hybrid models. While we developed improvements to sequence-to-sequence model training that closed the gap with hybrid models, the performance never surpassed hybrid models. This is in contrast to larger resource scenarios where hybrid models are consistently surpassed by sequence-to-sequence models. We believe additional focused effort on low-resource domain adaptation for sequence-to-sequence models could push these models past hybrid models.

One of the goals of MATERIAL was to explore the challenge presented by having a mismatch in style and domain between the training and test data. Indeed, MATERIAL gave us CTS data for training and the test was largely Broadcast speech. But, because typically CTS data is much more difficult to collect than Broadcast speech, it would have been more realistic to provide training data consisting primarily of Broadcast speech and evaluate on CTS data.

6.2 MT

Much effort was spent in this program searching for data for both supervised and semi-supervised training, and generally, more data improves system performance. While the component-level performance, such as MT, can be improved steadily over the span of a program, having access to a model that is pre-trained with orders of magnitude more data can completely change the research direction of a project. For example, at the most recent Principal Investigator meeting, the University of Southern California team showed that a 500 language-to-English MT system trained with 9 Billion words (Gowda 2021), when fine-tuned to MATERIAL languages, performed better than any single MT model one could have built from scratch. Another example is the rapid rise of very large generative pre-trained language models such as GPT-2 and GPT-3[11]. These pre-trained models can operate in the so called 1-shot or few-shot mode where only a few training data from a new domain are needed in order to give the state-of-the-art result on many traditional natural language processing tasks. It will be interesting to have a program designed around applying few-shot learning to complex problems such as those from MATERIAL where the performers' focus is on quick and novel adaptation of large pre-trained models.

6.3 CLIR

The success in finding data in the wild has not allowed us to understand the impact of very low resources on CLIR performance. This is because all of the languages in OP1 and OP2 were not actually low-resource languages. So, a follow-up research effort would concentrate on conditions with very little parallel data (for MT training) and very little transcribed data (for ASR training). Such a program would encourage researchers to develop unsupervised techniques for generating effective data representation or cross-language modeling. However, we would suggest that this research be performed – at least in part – on the same higher-resource languages used in MATERIAL in order to allow for comparison.

Retrieving plain text documents that satisfy an information need (expressed as a textual query) is only one mode of retrieval. A follow-up research effort could concentrate on different kinds of foreign documents, such as stylized documents (contracts, forms, records, etc.) and queries that

[11] https://en.wikipedia.org/wiki/GPT-3

ask questions related to the content of some of the fields in these forms. For example, in the case of mortgage documents, the information need could be "give me all documents in which the lender has imposed a mortgage rate that is at least 10% relative higher than the nationwide average."

6.4 Summarization

If we take "Summarization" to be an interaction between the system and the user, we can broaden the notion of interaction to be more useful by taking advantage of the system's ability to analyze a large corpus quickly. The usual interaction, where a user poses a query and the system returns documents that match the query is certainly useful. But this style of interaction often requires multiple queries as the user finds out what is in the corpus and narrows down their interest in light of the initial results. We think it is possible to have the system be an active participant in the satisfaction of a user's information need. For example, a request like "impact of COVID-19 in East Africa" is very broad, so the system can engage the user in a dialogue to narrow down the request. This can be accomplished by the system by first retrieving the large number of documents that are relevant to the initial query, clustering them along multiple dimensions, and then asking whether the user would like to narrow their query in one dimension or another, for example: the type of impact (military, political, civil, economic, health), the specific countries in East Africa (Sudan, South Sudan, Eritrea, Ethiopia, Somalia, Uganda, Kenya, Rwanda, Burundi, Tanzania), the time period (early or later in the pandemic), etc. Once the user has narrowed down one or more of the dimensions, the system can provide a more targeted response. By making the system an active participant, the user can more quickly find what they really need.

7.0 REFERENCES

ASR

G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan (2017), "An Exploration of Dropout with LSTMs," Interspeech, pp. 1586--1590, 2017.

R. Hsiao, R. Meermeier, T. Ng, Z. Huang, M. Jordan, E. Kan, *et al*., "Sage: The new BBN speech processing platform," Interspeech, pp. 3022-3026, 2016.

Y. Khassanov, S. Mussakhojayeva, A. Mirzakhmetov, A. Adiyev, M. Nurpeiissov, and H. Varol, "A Crowdsourced Open-Source Kazakh Speech Corpus and Initial Speech Recognition Baselinej," 2020. arXiv preprint arXiv:2009.10334.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, *et al.,* "The Kaldi Speech Recognition Toolkit," IEEE ASRU, 2011.

D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI," Interspeech, pp. 2751-2755, 2016.

Veselỳ, K., Ghoshal, A., Burget, L., & Povey, D., "Sequence-Discriminative Training of Deep Neural Networks," Interspeech, pp. 2345-2349, 2013.

MT

T. Gowda, Z. Zhang, C. A. Mattmann, and J. May, "Many-to-English Machine Translation Tools, Data, and Pretrained Models," 2021. https://arxiv.org/abs/2104.00290

M. Johnson, M. Schuster, Q. Le, M. Krikun, Y. Wu, Z. Chen, and N. Thorat "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," Transactions of the Association for Computational Linguistics, Volume 5, pp. 339-351, 2017.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," 2017. https://arxiv.org/abs/1706.03762

F. Wu, A. Fan, A. Baevski, Y. N. Dauphin and M. Auli, "Pay Less Attention with Lightweight and Dynamic Convolutions," In Proceedings of International Conference on Learning Representations, 2019.

L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz and S. Tsakalidis, "Enhancing Low Resource Keyword Spotting with Automatically Retrieved Web Documents," In Proceedings of Interspeech, Dresden, Germany, September 6-10, 2015.

CLIR

C. Callison-Burch, P. Koehn, and M Osborne, "Improved Statistical Machine Translation Using Paraphrases," In Proceedings of the Human Language Technology Conference of the NAACL, pp. 17-24, 2006.

D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen*, et al.,* "Score normalization and system combination for improved keyword spotting," In IEEE ASRU, pp. 210–215, 2013.

D. Karakos, R. Zbib, W. Hartmann, R. Schwartz, and J. Makhoul, "Reformulating Information Retrieval from Speech and Text as a Detection Problem," CLSSTS Workshop at LREC-2020.

J. Xu and R. Weischedel, "Cross-lingual Information Retrieval Using Hidden Markov Models," In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, ACL, Stroudsburg, PA, pp. 95–103, 2000.

R. Zbib, L. Zhao, D. Karakos, W. Hartmann, J. DeYoung, Z. Huang, Z. Jiang, N. Rivkin, L. Zhang, R. Schwartz, and J. Makhoul, "Neural-network lexical translation for cross-lingual IR from text and speech," In B. Piwowarski, *et al*., editors, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, pp. 645-654, July 21-25, 2019.

L. Zhang, D. Karakos, W. Hartmann, M. Srivastava, L. Tarlin, D. Akodes, S. Krishna Gouda, N. Bathool, L. Zhao, Z. Jiang, R. Schwartz, and J. Makhoul, "The 2019 BBN Cross-lingual Information Retrieval System," CLSSTS Workshop at LREC-2020.

Summarization

H. Lin and J. Bilmes, "A class of submodular functions for document summarization," In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 510–520, 2011.

J. Pennington, R. Socher, and C. D. Manning. 2014. "GloVe: Global Vectors for Word Representation," Stanford University, 2014. https://nlp.stanford.edu/projects/glove/

8.0 ACRONYMS

AQWV	Actual Query Weighted Value
AM	Acoustic Models
AMT	Amazon Mechanical Turk
ASR	Automatic Speech Recognition
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
BLEU	BiLingual Evaluation Understudy
BUT	Brno University of Technology
CLIR	Cross-Lingual Information Retrieval
CNet	Consensus Net
CNN-LSTM	Convolutional Neural Network-Long Short-Term Memory
CPU	Central Processing Unit
CTS	Conversational Telephone Speech
DARPA	Defense Advanced Research Projects Agency
Dev	Development
DNN-HMM	Deep Neural Network-Hidden Markov Model
E2E	End-to-End
EM	Expectation-Maximization
Eval	Evaluation
FA	False Accept
FLAIR	Foreign Language Automated Information Retrieval
FST	Finite State Transducer
GloVe	Global Vectors for Representation
GMM	Gaussian Mixture-Modeling
GPU	Graphics Processing Units
IARPA	Intelligence Advanced Research Projects Activity
IR	Information Retrieval
LF-MMI	Lattice-Free Maximum Mutual Information
LORELEI	Low Resource Languages for Emergent Incidents
LM	Language Model
M3S	Multi-Media Monitoring System
MAP	Mean Average Precision

MATERIAL	Machine Translation for English Retrieval of Information in Any Language
mBART	multilingual Bidirectional and Auto-Regressive Transformers
MFCC	Mel-Filter Cepstrum Coefficients
MIT	Massachusetts Institute of Technology
MQWV	Maximum Query Weighted Value
MT	Machine Translation
NB	News Broadcasts
NMT	Neural MT
NNJM	Neural Network Joint Model
NNLTM	Neural Network Lexical Translation Model
OOV	Out-Of-Vocabulary
OP1	Option Period 1
OP2	Option Period 2
OpenSLR	Open Speech and Language Resources
PBMT	Phrase-Based Hierarchical MT
Q1	Query set 1
QST	Query-Specific Thresholding
ReLU	Rectified Linear Unit
RNN-LM	Recurrent Neural Network Language Model
sMBR	state-level Minimum Bayes Risk
SMT	Statistical Machine Translation
SST	Semi-Tupervised Training
TB	Topical Broadcasts
TD	Truly relevant Documents
TDNN	Time Delay Neural Network
TER	Translation Edit Rate
TREC	Text REtrieval Conference
USCENTCOM	United States Central Command
USSOCOM	United States Special Operations Command
WER	Word Error Rate