Proceedings of the Cross-Language Search and Summarization of Text and Speech Workshop, pages 44–51
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

The 2019 BBN Cross-lingual Information Retrieval System

Le Zhang, Damianos Karakos, William Hartmann, Manaj Srivastava
Lee Tarlin, David Akodes, Sanjay Krishna Gouda, Numra Bathool, Lingjun Zhao
Zhuolin Jiang, Richard Schwartz, John Makhoul
Raytheon BBN Technologies
Cambridge MA, USA
{le.zhang,damianos.karakos,william.hartmann,manaj.srivastava}@raytheon.com
{lee.tarlin,david.akodes,sanjaykrishna.gouda,numra.saleem.ahmed.khan,lingjun.zhao}@raytheon.com
{zhuolin.jiang,rich.schwartz,john.makhoul}@raytheon.com

Abstract

In this paper, we describe a cross-lingual information retrieval (CLIR) system that, given a query in English, and a set of audio and text documents in a foreign language, can return a scored list of relevant documents, and present findings in a summary form in English. Foreign audio documents are first transcribed by a state-of-the-art pretrained multilingual speech recognition model that is fine tuned to the target language. For text documents, we use multiple multilingual neural machine translation (MT) models to achieve good translation results, especially for low/medium resource languages. The processed documents and queries are then scored using a probabilistic CLIR model that makes use of the probability of translation from GIZA translation tables and scores from a Neural Network Lexical Translation Model (NNLTM). Additionally, advanced score normalization, combination, and thresholding schemes are employed to maximize the Average Query Weighted Value (AQWV) scores. The CLIR output, together with multiple translation renderings, are selected and translated into English snippets via a summarization model. Our turnkey system is language agnostic and can be quickly trained for a new low-resource language in few days.

Keywords: cross-lingual informational retrieval, average query weighted value, AQWV

1. Introduction

The popularity of the Internet has made it easy to access vast amount of multilingual information for anyone. Yet, it is hard to understand information in a language you do not speak, not to mention searching through it. Cross-Language Information Retrieval (CLIR) and Summarization make it possible to break the language barrier and to make domain information accessible to all users irrespective of language and region.

The IARPA MATERIAL[1] program presents us with the challenge of developing high-performance CLIR, machine translation, automatic speech recognition (ASR), and summarization for a new language in a few weeks, given limited training resources. In this paper, we describe our CLIR system entry to the MATERIAL evaluation of October, 2019. We were to process evaluation data for both Lithuanian and Bulgarian and to submit system output in 10 days.

Our CLIR system achieves the same goal as the SARAL system (Boschee et al., 2019a). While both systems feature a neural network (NN) architecture, the main difference lies in the way an NN model is used. The SARAL system uses a neural network attention model (dot-product) to compute query-document relevance from a shared embedding space, while our system utilizes neural network (multilayer perceptron) as part of the Neural Network Lexical Translation Model (Zbib et al., 2019) to produce probability of translation needed by a probabilistic CLIR model.

The rest of this paper is organized as follows: we introduce the task and data in section 2, including a high level overview of the technical approach. Subsequent sections describe each individual component of the system in more detail. Section 3 and 4 cover Automatic Speech Recognition and Machine translation, two of the key pre-processing components. The CLIR component is presented in section 5 while Summarization is described in section 6. We present the result in section 7 and discuss the application of the system to low resource languages in section 8. We conclude this paper in section 9.

2. Task and Data

The task of MATERIAL evaluation is given a set of foreign language documents and English queries, retrieve documents that are relevant to each query and generate a summary in English for each document the system deems relevant to the query. Note that the MATERIAL summaries are query-biased, i.e. the purpose of a summary is to allow an English speaker to judge whether the original foreign language document might have been relevant to the query. It is query-biased summary of thoughts not general document summary.

2.1. The AQWV Metric

The main evaluation metric is the Average Query Weighted Value score, a numerical score for every query-document pair, and is defined as a linear combination of the miss and false alarm rates:

$$AQWV = 1 - \left(\overline{\text{pMiss}} + \beta\ \overline{\text{pFA}}\right) \tag{1}$$

$\overline{\text{pMiss}}$ is the average per-query miss rate and is defined as follows

$$\overline{\text{pMiss}} = \frac{1}{|Q_r|} \sum_{q \in Q_r} \frac{\#\text{ misses of } q}{\#\text{ refs of } q} \tag{2}$$

where Q_r is the set of queries with references in the data (i.e., each query has at least one relevant document). The number of references and the number of misses of query

[1] https://www.iarpa.gov/index.php/research-programs/material

q is computed based on the whole document collection $\mathcal{C}$ under consideration.

$\overline{\text{pFA}}$ is the average per-query false alarm rate and is defined as follows

$$\overline{\text{pFA}} = \frac{1}{|Q|} \sum_{q \in Q} \frac{\text{\# FAs of } q}{|\mathcal{C}| \text{ - \# refs of } q} \tag{3}$$

The constant β in Equation (1) reflects the relative importance of the two types of error.

One can also compute a per-query performance measure, the Query Weighted Value (QWV), defined for query q as

$$QWV = 1 - \text{pMiss}(q) - \beta \text{ pFA}(q) \tag{4}$$

The AQWV metric has several important properties. The range is $(-\beta, 1]$, where a system that returns no detections would obtain a score of 0. It is possible for a system with a large number of false alarms to give a negative score. A correct detection for different queries is not weighted equally—the gain is related to the rarity of the query, as queries with fewer relevant documents gain more from each correct detection (e.g., think of the case where a query has only one truly relevant document, and, assuming no false alarms, accepting/rejecting that document will result in a QWV of one/zero). If we ignore the number of true references for a query in Equation 3—often this is reasonable as the number of documents dwarfs the number of true references—then there is a constant penalty for every false alarm. The constant β controls the strength of the penalty. All results in this paper use a β of 40, required by the evaluation task. This means the system has to be tuned to produce a very low false alarm rate: a single false alarm is penalized 40 times more than a single true miss. The general idea behind a high value of β is to minimize the amount of non-relevant documents the end user has to look through when using a CLIR system. The evaluation plan also suggests an effective CLIR system should reach an AQWV value of 0.5 or higher.

In the rest of the paper, we will denote by MQWV the maximum value that AQWV can attain by sweeping over all possible decision thresholds.

2.2. Data

The training dataset (Build set) consists of approximately 50 hours of audio (conversational telephone speech) for ASR and 800k words of bitext for MT. There are additional Dev and Analysis datasets drawn from the same data pool as the Evaluation dataset for internal testing and error analysis purpose.

Our system will be evaluated on the blind Evaluation dataset, which is not guaranteed to have the same query relevance probability as that of the Dev or Analysis dataset. Table 1 gives the size of each dataset we received. We also used existing additional speech and parallel text for building multilingual ASR and MT models. The detailed data used by each component will be covered in individual sections below.

2.3. Technical Approach

Figure 1 is a top-level block diagram of our CLIR and Summarization system. More details about the various components appear in Sections 5 and 6. At top right is an existing corpus of foreign audio and text documents, which go through ASR, translation and indexing steps. At top left, a user issues a query in English, which is then expanded through query expansion. Note term translations for CLIR can happen either after query expansion (from English to the foreign language) or in document precomputation (from the foreign language to English). Our preferred mode is to efficiently translate all terms in the foreign documents in all possible ways using the context of nearby words. Note that, given that documents have a longer context than queries, translation of documents to English is more precise than translation of short (nominally one-two words) English queries to the foreign language. The preprocessed data in the form of weighted search terms (from query expansion) and indexed documents serves as input to the CLIR query module, where each document receives a query relevance score. This is followed by score normalization, combination, and thresholding to maximize AQWV scores on the Dev or Analysis set. Finally, the retrieved documents, together with translation models and target in-

	Lithuanian		Bulgarian	
Dataset	Text	Speech	Text	Speech
Build	610K	66 hr	735K	41 hr
Dev	174K	10 hr	202K	15 hr
Analysis	234K	10 hr	276K	18 hr
Evaluation	4.3M	172 hr	4.5M	183 hr

Table 1: Size of text (number of source tokens) and speech data provided in each language pack.

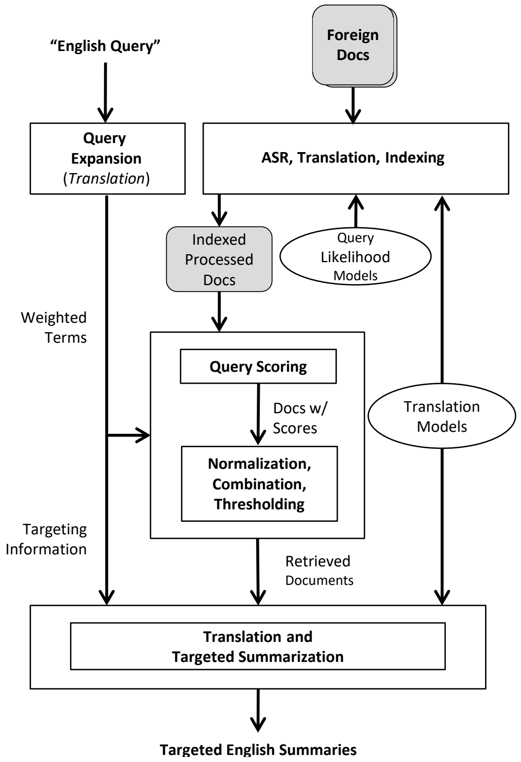


Figure 1: Block diagram of our CLIR and Summarization system

formation (where the CLIR evidence is from in the MT target), go through the Translation and Summarization module to produce final summary snippets in English.

3. Automatic Speech Recognition

3.1. Training Data

Our acoustic model is pretrained on approximately 1500 hours of narrowband conversational data from 11 languages (Keith et al., 2018). This pretrained multilingual model is then fine tuned to the target language. The fine tuning data consisted of the build_train and build_dev portions of the MATERIAL data. Note that, although the provided transcribed training speech is conversational telephone speech, the majority of the evaluation speech is wideband broadcast speech. So we also collected approximately 700 hours of untranscribed wideband data from YouTube in both Bulgarian and Lithuanian for semi-supervised training. We expand the acoustic training data by creating two additional copies that are augmented by noise, compression, and reverberation (Hartmann et al., 2016).

The language models are trained using four data sources: 1) acoustic transcripts, 2) build_bitext, 3) paracrawl (Esplà et al., 2019), 4) automatically collected web data. The procedure used to collect web data is described in (Zhang et al., 2015). A separate trigram language model is created for each data source and then interpolated to create a single language model.

3.2. Pronunciation Lexicon

We start with the original pronunciation lexicon provided with the build data. These pronunciations are also used to train a model using SequiturG2P (Bisani and Ney, 2008) in order to generate pronunciations for any additional words. Our final lexicon contains all words from the original build lexicon, the build_bitext, and the paracrawl data. We also include the most frequent 300k words in the web data. Combined, this brings the total number of words in the lexicon to approximately 400k.

3.3. Acoustic Model

We use a CNN-LSTM acoustic model. This model is similar to the TDNN-LSTM acoustic model, but with 8 convolutional layers prepended. In addition to the standard melfilterbank features, we also include i-vectors for speaker adaptation. The Sage toolkit (Hsiao et al., 2016) is used for training and decoding with acoustic model training based on the Kaldi Chain model. Training consists of a single epoch using the LF-MMI criterion followed by an additional epoch using sMBR. After the supervised model is trained, we perform semi-supervised training. The original model is used to transcribe the collected YouTube data. We combine this automatically transcribed data with the original labeled data and retrain the model. Note that while both the supervised and unsupervised data are used during LF-MMI training, only the supervised data is used during sMBR training in order to limit the effects of errors in the unsupervised transcripts.

3.4. Language Model

We build both n-gram and RNN-based language models (LM). A trigram LM is constructed from each of the four sources of text data. The LMs are then interpolated with weights that minimize perplexity on the Analysis data. The RNN-LM is trained on the same set of data as described in (Xu et al., 2018). The neural model consists of two LSTM layers and three fully connected layers.

3.5. Decoding

All audio data is first decoded using the above described acoustic model with a trigram language model to generate initial lattices. The lattices are then rescored using the RNN-LM. The final step is to convert the rescored lattices into confusion networks (CNets).

4. Machine Translation

4.1. Training Data

The primary data source for constructing MT models is parallel data from the build language pack, augmented with a variety of web data, such as CommonCrawl[2] and open parallel corpus (Tiedemann, 2012). We used the PanLex dictionary (Kamholz et al., 2014) for the languages, simply by treating each translation as a (very short) parallel sentence. We also used parallel data from Russian and Ukrainian for building multilingual neural MT models. We employed an oversampling technique to ensure that the target languages (Lithuanian and Bulgarian) are well represented in the training. More specifically, we oversample data from each language with different oversampling factors so that the target language has a proportion of 70% in the final training data, while the other three languages have an equal proportion of 10% each. Table 2 summarizes the amount of training data used:

Language	Source Tokens (millions)
Lithuanian	13.1
Bulgarian	20.5
Russian	14.0
Ukrainian	9.7

Table 2: Amount of parallel data used in training multilingual MT models.

Our system needs to translate both the text data and the transcript from the ASR sub-system for use with summarization (Section 6). Because there is no casing information in the ASR transcript, we augmented the MT training data with the lower-cased version of the source data with punctuation marks removed to mimic the condition of ASR output. The neural MT models were trained on both versions of the data together, in a single "multi-style" fashion, to handle both text and ASR transcript as input. This was however not done for the phrase-based model described below.

[2] http://commoncrawl.org

4.2. MT Models

The machine translation component consists of two multilingual neural MT models and one phrase-based statistical MT (SMT) model:

1. Transformer NMT: a 6-layer transfomer-based model (Vaswani et al., 2017) jointly trained over Lithuanian, Bulgarian, Russian and Ukrainian data. We applied data oversampling and used 21k subword units in the vocabulary. We trained the model over the training data using 600k training steps with a batch size of 2048. We averaged the last 3 checkpoints to produce the final model.

2. DynamicConv NMT: a 6-layer dynamic convolution model (Wu et al., 2019) trained over the same data with 50k subword units. 1200k updates were used for the training. The final model was produced by model averaging of the last 3 checkpoints.

3. Moses Phrase-based SMT: a phrase-based statistical MT system trained over the Lithuanian or Bulgarian bilingual data.

All MT models produce N-best (N=20) hypotheses as output for downstream summarization processing. We used the tensor2tensor toolkit (Vaswani et al., 2018) for the transformer implementation and the fairseq toolkit (Ott et al., 2019) for the dynamic convolution model. We also used Moses (Koehn et al., 2007) for training the phrase-based model. Our own tokenizer was used instead of the tokenizer from Moses to match the tokenization scheme used by other system components. Subword tokenization was done using the sentencepiece toolkit (Kudo and Richardson, 2018), an unsupervised text tokenization method that is independent of the language being processed.

5. CLIR

The CLIR system consists of a number of components for performing indexing, query processing, retrieval, score normalization, system combination, and thresholding. These components are described in more detail below.

5.1. Query Processing

We treat queries in two distinct ways: (i) as *flat strings*, where the query words are used as a "bag of words", completely ignoring the context-free nature of the queries; this is the mode used in the paper (Zbib et al., 2019); (ii) as *hierarchical*, expressed using a parse tree, where the MATERIAL-provided context-free grammar (CFG) is used for this purpose. The leaves of the tree correspond to individual terms, while internal nodes of the tree correspond to various query types such as *LEXICAL PHRASE*, *PLUS*, *EXAMPLE_OF*, etc.

In the case of the *flat* query treatment, we consider query translation (to the foreign language) as well as document translation (to English) as two distinct modes of retrieval.

In the case of the *parse tree*, PLUS and EXAMPLE_OF (CONCEPTUAL) query components are further *expanded* to include additional query terms that are used in the search. Specifically, the terms of the PLUS components are expanded using nearest-neighbor words of English pre-trained Wikipedia-derived word embeddings (Bojanowski et al., 2017) (with a minimum cosine similarity $\cos_{\min}$, typically between 0.3-0.4). The weight of each expansion is an exponential function of the cosine similarity, as follows

$$w = \exp\{-\alpha(1 - \cos)/(1 - \cos_{\min})\} \tag{5}$$

where α is a tunable coefficient (typically equal to 3.0 in our experiments). This weight is multiplied with the probability of occurrence of the term in the document. The terms of the EXAMPLE_OF components are expanded using both WordNet and pre-trained monolingual embeddings as follows:

- Pre-Processing: Find all senses of the EXAMPLE_OF argument phrase in WordNet as NLTK Synset objects.

- WordNet Hyponym Traversal: For each Synset, recursively traverse its hyponym tree and record all hyponyms found during the process.

- Post-Processing: Filter out any hyponyms that have a vector cosine distance relative to the original EXAMPLE_OF phrase greater than 0.35. As above, we use the word embeddings from (Bojanowski et al., 2017).

For instance, the expansions for the query EXAMPLE_OF(footwear) include: "baby shoe", "bowling shoe", "sneaker", "wooden shoe", "rubber boot", "congress shoe", "ghillie", "combat boot", "footgear", "huarache", etc.

5.2. Indexing

We construct inverted indexes for both the source language and the target language. For text documents, we index words and n-grams. For speech documents, we index both the 1-best output (which is treated as regular text) and the confusion network, saving the ASR posterior score. The index contains the location of the words and the n-grams as well as the probability of translation to the target (query) language, scaled by the ASR posterior in the case of speech. The probability of translation is obtained from the GIZA translation table (generating GIZA alignments is usually one of the first steps run in a MT system), interpolated with the Neural Network Lexical Translation Model (NNLTM) score. More details about NNLTM can be found in (Zbib et al., 2019). Note that the indexing is done with both original and stemmed English words.

5.3. Retrieval Models

The individual retrieval models are as follows:

- For "flat" queries, four retrievals are performed: with original/stemmed words and with document/query translation. (Obviously, the appropriate index is used in each case.) For the case of document translation, two confidence score computations are also done: using the simple probabilistic model and with the probability of occurrence (see (Zbib et al., 2019) for details).

- For "hierarchical" queries, the parse tree is used as a "processing tree", akin to an *abstract syntax tree* used in computer language compilers. Then, the process of retrieval can be accomplished using a depth-first traversal of the tree. Terminal nodes compute the locations where individual query words (original or expanded) are matched in a document, based on what is in the inverted index. Internal (non-terminal) nodes of the tree perform an operation corresponding to the query type: e.g., for PLUS or EXAMPLE_OF queries, the individual retrievals of the "children" nodes (query terms or expanded terms) are combined using a probability of occurrence operation. Similarly, an internal node that corresponds to a LEXICAL PHRASE only keeps retrievals that are "close" to each other and penalizes for missing phrase words.

- Whole-phrase matching, where, if a phrase query component existed in the phrase translation table, and if the source phrase translation existed in the document, the corresponding probability was used in the retrieval.

- In all cases above, two retrievals are done in the case of Speech: using ASR 1-best and ASR confusion networks (cnets). While the cnets provide better performance, the 1-best helps in combination.

5.4. Normalization, Combination, and Thresholding

The detection scores of each of the individual systems are normalized using a learned model. The model computes a linear combination of the following features:

1. Original retrieval $\mathit{score}(q,d)$ for query q and document d

2. The QST-transformed score $\mathit{score}_{\text{qst}}(q,d)$, where QST is a technique similar to KST, described in (Karakos et al., 2013)

3. The normalized sum $\sum_{d \in C} \mathit{score}(q,d)/|\mathcal{C}|$

4. The three features:

   $$\min_{w \in q}\{\mathit{score}(w,d)\},\ \max_{w \in q}\{\mathit{score}(w,d)\},\ \operatorname{avg}_{w \in q}\{\mathit{score}(w,d)\},$$

   where avg_w is just the average over all words w in query q (esp. for multi-word queries).

5. The three features:

   $$\min_{w \in q}\{\text{count}(w)\},\ \max_{w \in q}\{\text{count}(w)\},\ \operatorname{avg}_{w \in q}\{\text{count}(w)\},$$

   where count(w) is the count of w in the IR training data (e.g., parallel data used to train the bilingual dictionary for CLIR).

The weights in the linear combination are computed using Powell's method (Karakos et al., 2013), with the objective to maximize MQWV.

Combination of a subset of the individual systems (determined through performance on Analysis and Dev) is done by interpolating the log probabilities from the different systems, with weights determined using Powell's method, as mentioned above.

The final output on the test set is thresholded by tuning the overall proportion of accepted documents according to performance of the query set on the Analysis document set.

5.5. Evidence for Summarization

Besides outputting scores and decisions for all documents that have been accepted, the CLIR system outputs an evidence "object" for each sentence that has a nonzero score for a query. The evidence object specifies the source segment, source word, query word found, and the probability for that query word. These evidence objects are just referred to as "evidence" in the summarization section below.

6. Summarization

6.1. Overview

The task of the summarization component is to create English-language summaries for the documents that are retrieved by the CLIR component. The summarization component makes use of query "evidence" provided by CLIR component and English translations provided by the MT component to rank and select appropriate sentences (or fixed-length snippets) in order to form a summary that can be presented to human users. Below we describe in more detail, the mechanism to use output from CLIR and MT components, our extractive selection algorithm, and some presentation aspects of the summarization component.

6.2. Combining Output from Multiple CLIR Systems

As explained above, the CLIR component is comprised of multiple systems that each produce their individual output. While the system combination step in CLIR takes care of combining the relevance decision and document-level relevance scores output by these systems, the word-level evidence information is combined by the summarization component. This combined information is then used in the sentence selection process (described below). The word-level evidence provides, for every query word likely to appear in a sentence, the probability of its occurrence. This probability is derived by interpolation of GIZA and NNLTM translation probabilities. The summarization component uses a weighted sum of these probabilities to form an aggregate score for a query word appearing in a sentence.

6.3. Combining Output from Multiple MT Systems

The summarization component uses top-K English sentences from the nbest output of each of the three MT systems–Transformer, DynamicConv, and Moses. For the evaluation, the value of K was set to 4. Summarization component looks for specific query words within these sentences based on the evidence provided by CLIR and also a direct string match. It then creates fix-sized snippets around these query words. These snippets are then used for ranking and selection to form the final summary.

Note that the summarization component has the ability to extract either full sentences or fix-sized snippets in order

Language	Text			Speech			All
	AQWV	pMiss	pFA	AQWV	pMiss	pFA	AQWV
Lithuanian	0.617	0.287	0.002	0.609	0.306	0.002	**0.613**
Bulgarian	0.695	0.186	0.003	0.654	0.210	0.003	**0.675**

Table 3: Official AQWV scores for Text and Speech data on the evaluation set with a β of 40. The All column reports a single AQWV system score computed as the mean of the Text and Speech AQWV scores.

to create the summary. For the evaluation, we chose to use fix-sized snippets that extend up to 7 words before and after the query word.

6.4. Snippet Selection Algorithm

Our extractive snippet selection algorithm is a submodular selection algorithm that uses both query-evidence and tf-idf scores in its coverage and diversity objectives (Lin and Bilmes, 2011). The query words that are discovered by direct string match, and that for some reason were not captured in interpolated GIZA-NNLTM translation tables, are assigned a fixed score. It is also worth mentioning that we do some special handling for expanded queries. For expanded queries, we use a cutoff on the list of expanded query terms, so as to reduce possible noise in summary output and also lower the computation time needed for snippet-selection itself. We experimented with various cutoffs and found that a cutoff of 3 worked best for text summaries, while a cutoff of 0 (no expanded terms) worked best for audio summaries.

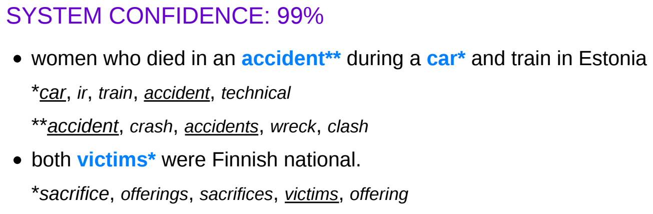
We select the top two snippets ranked by the submodular algorithm to form the final summary, which in part is motivated by (Maxwell et al., 2017), who show that for a summarization system for IR, longer summaries are not necessarily beneficial for human-in-the-loop relevance judgments. Since we use nbest English sentences from multiple MT systems, there is a possibility (although bleak) that some adjacently ranked snippets can have a large information overlap. To address that, after selecting a snippet from a given unique sentence (based on the mapping sentence ID from the foreign language side), we preclude other snippets from that sentence from the selection process.

6.5. Presentation Aspects of Summaries

Based on the presentation scheme used by (Boschee et al., 2019b), our summaries have the query words (or any word that is likely to be an alternative translation for the query word) highlighted in blue. A footnote is also attached to each highlighted word, which is composed of the alternative translations that the highlighted word could have in the context. These alternative translations are the top 5 words appearing in a combined GIZA-NNLTM interpolated translation table, where the combined table is created by applying Borda ranking[3] to multiple GIZA-NNLTM interpolation tables used by various CLIR systems. See figure 2 for a sample summary from the Lithuanian system.

7. Results

Table 3 gives the official AQWV scores for Lithuanian and Bulgarian on Text and Speech conditions of the evaluation data. A β of 40 is used to penalize false alarms when computing AQWV scores. Consequently our system is tuned to produce a very low probability of the average per-query false alarm ($\overline{\text{pFA}}$) at the cost of relatively high probability of miss ($\overline{\text{pMiss}}$).



SYSTEM CONFIDENCE: 99%

- women who died in an **accident**** during a **car*** and train in Estonia
  *car, ir, train, accident, technical
  **accident, crash, accidents, wreck, clash
- both **victims*** were Finnish national.
  *sacrifice, offerings, sacrifices, victims, offering

Figure 2: Sample summary snippet returned from the Lithuanian system for a plus query "car accident victim"+

In addition to the AQWV results on the evaluation set, we also present results from our ASR and MT components on the Analysis set where we have references. In table 4 we give the word error rate (WER) and BLEU scores our system produced on the Analysis sets. The BLEU scores are obtained using the `mteval-v11b` scoring script from NIST.[4] For the MT result, the two neural MT models, transformer (NMT-T) and dynamic convolution (NMT-D), have similar performance, and are much better than that from the SMT Moses system. The gain of the NMT over SMT model is largely due to multilingual training, which is not possible with the phrase based SMT. Because sometimes our Summarization component will choose the rendering of a snippet from the SMT instead of that from the neural MT system, we decide to include SMT as part of the translation pipeline.

Language	WER	BLEU		
MT Model		NMT-T	NMT-D	SMT
Lithuanian	18.7	30.4	30.5	20.0
Bulgarian	17.6	43.8	43.5	34.7

Table 4: WER and BLEU scores for Lithuanian and Bulgarian on the Analysis set.

8. Low-Resource Languages

In this paper, we only reported experimental results from two medium-resource languages as part of the October 2019 MATERIAL evaluation. However, all the techniques discussed in this paper are applicable to low-resource languages. Since such languages have very limited training data, techniques such as semi-supervised training, can be

[3] http://en.wikipedia.org/wiki/Borda_count

[4] https://www.nist.gov/itl/iad/mig/tools

employed to leverage large amounts of existing or web collected data to further improve system performance. This can be done for speech recognition or machine translation via back translation (Sennrich et al., 2016). Previously, we had applied our system to low-resource languages such as Somali, Swahili, and Tagalog. More recently, we applied our system to Pashto as part of the MATERIAL Surprise language Sprint in early 2020 and achieved very good performance.

9. Summary

In this paper, we presented a CLIR system that can perform information retrieval over audio and text documents from a foreign language and present summaries in English. Key features of our system include an appropriate probabilistic CLIR model that uses a neural network lexical translation model, strong multilingual neural speech recognition and neural translation models, plus advanced score normalization, combination, and thresholding schemes. Furthermore, our system is language agnostic and can be quickly brought up for a new low-resource language in a few days. In the future, we plan to explore better ways of using harvested data to enhance CLIR, ASR, and MT in the form of semi-supervised training.

10. Acknowledgements

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Air Force Research Laboratory contract number FA8650-17-C-9118.

11. Bibliographical References

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434 – 451.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boschee, E., Barry, J., Billa, J., Freedman, M., Gowda, T., Lignos, C., Palen-Michel, C., Pust, M., Khonglah, B. K., Madikeri, S., May, J., and Miller, S. (2019a). SARAL: A low-resource cross-lingual domain-focused information retrieval system for effective rapid document triage. In Marta R. Costa-jussà et al., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 19–24. Association for Computational Linguistics.

Boschee, E., Barry, J., Billa, J., Freedman, M., Gowda, T., Lignos, C., Palen-Michel, C., Pust, M., Khonglah, B. K., Madikeri, S., May, J., and Miller, S. (2019b). SARAL: A low-resource cross-lingual domain-focused information retrieval system for effective rapid document triage. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24, Florence, Italy, July. Association for Computational Linguistics.

Esplà, M., Forcada, M., Ramírez-Sánchez, G., and Hoang, H. (2019). ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland, August. European Association for Machine Translation.

Hartmann, W., Ng, T., Hsiao, R., Tsakalidis, S., and Schwartz, R. M. (2016). Two-stage data augmentation for low-resourced speech recognition. In Nelson Morgan, editor, *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 2378–2382. ISCA.

Hsiao, R., Meermeier, R., Ng, T., Huang, Z., Jordan, M., Kan, E., Alumäe, T., Silovský, J., Hartmann, W., Keith, F., Lang, O., Siu, M., and Kimball, O. (2016). Sage: The new BBN speech processing platform. In Nelson Morgan, editor, *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 3022–3026. ISCA.

Kamholz, D., Pool, J., and Colowick, S. M. (2014). Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 3145–3150.

Karakos, D., Schwartz, R. M., Tsakalidis, S., Zhang, L., Ranjan, S., Ng, T., Hsiao, R., Saikumar, G., Bulyko, I., Nguyen, L., Makhoul, J., Grézl, F., Hannemann, M., Karafiát, M., Szöke, I., Veselý, K., Lamel, L., and Le, V. B. (2013). Score normalization and system combination for improved keyword spotting. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, pages 210–215. IEEE.

Keith, F., Hartmann, W., Siu, M., Ma, J. Z., and Kimball, O. (2018). Optimizing multilingual knowledge transfer for time-delay neural networks with low-rank factorization. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4924–4928. IEEE.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

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

Maxwell, D., Azzopardi, L., and Moshfeghi, Y. (2017). A study of snippet length and informativeness: Behaviour, performance and user experience. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 135–144, New York, NY, USA. Association for Computing Machinery.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2214–2218.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, L., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., and Uszkoreit, J. (2018). Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, AMTA 2018, Boston, MA, USA, March 17-21, 2018 - Volume 1: Research Papers*, pages 193–199.

Wu, F., Fan, A., Baevski, A., Dauphin, Y., and Auli, M. (2019). Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Xu, H., Chen, T., Gao, D., Wang, Y., Li, K., Goel, N., Carmiel, Y., Povey, D., and Khudanpur, S. (2018). A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5929–5933. IEEE.

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

Zhang, L., Karakos, D., Hartmann, W., Hsiao, R., Schwartz, R. M., and Tsakalidis, S. (2015). Enhancing low resource keyword spotting with automatically retrieved web documents. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 839–843. ISCA.