ELISA System Description for LoReHLT 2018

Ronald Cardenas, Thamme Gowda, Ulf Hermjakob, Nima Pourdamghani, Michael Pust, Jibiao Shen, Tian Xie, Jonathan May
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
jonmay@isi.edu

Nikolaos Malandrakis, Ruchir Travadi, Victor Martinez, Karan Singla, Colin Vaz, Shrikanth Narayanan
Viterbi School of Engineering
University of Southern California
Los Angeles, CA 90089
shri@sipi.usc.edu

Kenton Murray, Toan Nguyen, David Chiang
Dept. of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
dchiang@nd.edu

Boliang Zhang, Xiaoman Pan, Ying Lin, Di Lu, Lifu Huang, Tongtao Zhang, Kevin Blissett, Ni Zhang, Spencer Whitehead, Ananya Subburathinam, Diya Li, Qingyun Wang, Zhiying Jiang, Heng Ji
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180
jih@rpi.edu

Ondrej Glembek, Murali Karthick Baskar, Santosh Kesiraju, Lukas Burget, Karel Benes, Igor Szoke, Karel Vesely, Jan "Honza" Cernocky
Dept. of Computer Graphics and Multimedia
FIT, Brno University of Technology
61266 Brno, Czech Republic
glembek@fit.vutbr.cz

Derry Wijaya and Chris Callison-Burch
University of Pennsylvania
Philadelphia, PA 19104
ccb@cis.upenn.edu

Hongzhi Xu, Charles Yang, Mitch Marcus
University of Pennsylvania
Philadelphia, PA 19104
mitch@cis.upenn.edu

Brian Moran, Leanne Rolston, Gina-Anne Levow
Department of Linguistics
University of Washington
Seattle, WA 98195
levow@uw.edu

Wenda Chen, Camille Goudeseune, Leda Sari, Mark Hasegawa-Johnson
Beckman Institute for Advanced Science and Technology
University of Illinois
Urbana, IL, 61801
levow@uw.edu

I. Tasks, Conditions and Submissions

We participated in all four tasks: Entity Discovery and Linking (EDL), Machine Translation (MT), Situation Frame from text(SFT), and SF for speech (SFS). In all tasks we generally submitted to the constrained track; where submissions were to the unconstrained track, this may be regarded as another constrained track submission, either mis-submitted or taking advantage of limited submission space, if relevant.

A. EDL Highlights

Some new and successful approaches for EDL include:

- We developed a novel approach to construct multi-lingual common semantic space [1] and use it for multi-lingual multi-task transfer learning [2] which allows related languages to share and transfer resources and knowledge. For this evaluation we used Swahili as the related language for Kinyarwanda and Bengali as the related language for Sinhala .
- We developed various incident-drive collective inference methods for entity linking to Geoname database.
- We developed a novel cross-lingual joint entity and word embedding approach for parallel sentence mining and name translation mining, which are used for improving Machine Translation and Speech Recognition.

B. MT Highlights

We used a variety of traditional MT systems, including phrase-based, Hiero, and syntax-based approaches. We also used a variety of neural MT systems, both sequence-to-sequence systems (one of which is specifically oriented for low-resource translation) and tensor-to-tensor systems (one of which uses back-translations of the Leidos Reliefweb corpus). Some approaches that worked in previous years (using related language data as incident data, building special out-of-vocabulary (OOV) word translators, building a do-not-translate tagger) were unnecessary or low priority in this year, and other approaches (neural MT, selection from comparable corpora, system combination) that were not helpful previously were helpful this year. As always, finding large amounts of clean data is paramount. The solution this year was to re-align the provided parallel data, which was given with bad sentence alignments. By treating this data as generally document-aligned and comparable but not sentence-aligned or necessarily parallel, we were able to generate a reasonably clean parallel sentence set suitable for cleaning. Additionally, we found the Native Informants (particularly for Sinhala) quite good at translation this year and were able to effectively develop in-domain parallel data. We were also able to create pseudo-parallel data with our Chinese Room interface but not in enough time to interface with NIs.

C. Text SF Highlights

We integrated Status variable generation into the SF Type systems, using a multi-task learning approach, where each Type and status variable are treated as separate tasks, enabling the use of data with missing labels. We augmented our training data by, apart from all the released Situation Frame development datasets, using the released speech SF development datasets, after transcription through Automatic Speech Recognition and translation using the ELISA, Google and Bing Machine Translation interfaces. We used a dictionary-based hashtag and twitter handle splitter as a pre-processing step, that allowed our SF models to understand hashtags and handles.

D. Speech SF Highlights

We were able to build 24-hour systems, which, for our pipeline, require ASR, Information Extraction, Machine Translation, **and** SF identification, largely due to the ability to recognize speech in only 2-3 hours from receiving data. We did this using UIUC's new high-speed speech recognizer ASR24. Similar to the SF text task, we used a multi-task learning approach where a shared model was used for predicting Type and status variables. We also augmented both SF-annotated text as well as ASR-transcribed and translated speech documents for training the models. The multi-task nature of learning and augmentation of text documents in the training data were both crucial especially for urgency prediction, since most of the speech datasets lacked urgency annotation.

II. Entity Discovery and Linking

Our Entity Discovery and Linking (EDL) team consists of Boliang Zhang, Xiaoman Pan, Ying Lin, Di Lu, Lifu Huang, Tongtao Zhang, Kevin Blissett, Ni Zhang, Spencer Whitehead, Ananya Subburathinam, Diya Li, Qingyun Wang, Zhiying Jiang and Heng Ji.

A. Submissions

For the EDL task we only participated in constraint setting for both check points. Tables I summarizes the submissions for each checkpoint.

B. Core Algorithmic Approach

The overall framework follows our cross-lingual EDL system for 282 languages [3], [4] and consists of three steps: (1) Incident Language (IL) name tagging and English name tagging; (2) Translate IL names to English and link English mentions to English knowledge base (KB); and (3) cluster unlinkable (NIL) name mentions across IL and English. We will present detailed approach for each step as follows.

Name Tagging. We use a typical neural network architecture that consists of Bi-directional Long Short-Term Memory and Conditional Random Fields network [5] as our underlying learning model for name tagging. We acquire training data through our Chinese Room annotation interface [6] which allows an English speaker to annotate names for any language. We annotated 1,889 Kinyarwanda sentences and 2,438 Sinhala sentences for check point 1; and 7,900 Kinyarwanda sentences and 5,297 Sinhala sentences for check point 2. We made the following novel additions and new techniques this year.

Multi-lingual common space and cross-lingual transfer learning. We construct a multilingual common semantic space [1] based on distributional semantics, where words from multiple languages are projected into a shared space to enable knowledge and resource transfer across languages. Beyond word alignment, we introduce multiple cluster-level alignments and enforce the word clusters to be consistently distributed across multiple languages. We exploit three signals for clustering: (1) neighbor words in the monolingual word embedding space; (2) character-level information; and (3) linguistic properties (e.g., apposition, locative suffix) derived from linguistic structure knowledge bases available for thousands of languages. We introduce a new cluster-consistent correlational neural network to construct the common semantic space by aligning words as well as clusters. Intrinsic evaluation on monolingual and multilingual QVEC tasks shows our approach achieves significantly higher correlation with linguistic features than state-of-the-art multi-lingual embedding learning methods do. We then feed the multi-lingual embedding representation into a novel cross-lingual transfer learning framework [2] so we can use Swahili name tagging training data for Kinyarwanda and Bengali training data for Sinhala .

Global Attention. Many name tagging approaches use local contextual information with much success, but fail when the local context is ambiguous or limited. We developed a

TABLE I
ELISA Kinyarwanda and Sinhala EDL Submissions

Check Point	Condition	Submission (Kinyarwanda /Sinhala)	Description
1	Constrained	356/357	Full system trained on RPI 6 hours Chinese Room data.
1	Constrained	487/489	Full system + cross-lingual transfer learning from related languages.
1	Constrained	545/547	Full system - hashtag and twitter user ID processing.
1	Constrained	511/509	Full system trained on RPI 10 hours Chinese Room data.
1	Constrained	488/490	Full system trained on RPI + JHU Chinese Room data.
2	Constrained	605/606	Full system trained on RPI 5 days Chinese Room data.
2	Constrained	673/675	Full system - GPE designator post-processing.
2	Constrained	700/704	Full system - hashtag and twitter user ID processing.
2	Constrained	699/703	Full system - ensemble learning
2	Constrained	769/782	Full system trained on RPI + JHU Chinese Room data.
2	Constrained	775/785	Full system trained on RPI adjudicated data only.
2	Constrained	779/862	Full system - nominal extraction.
2	Constrained	820/789	Full system trained on RPI 7 days Chinese Room data.
2	Constrained	672/674	Full system + cross-lingual transfer learning from related languages.
2	Constrained	607/608	Full system + cross-lingual transfer learning based on incident-related document selection from related languages.

new framework to improve name tagging by utilizing local, document-level, and corpus-level contextual information. We retrieve document-level context from other sentences within the same document and corpus-level context from sentences in other documents. This model learns to incorporate document-level and corpus-level contextual information alongside local contextual information via document-level as well as corpus-level attentions, which dynamically weight their respective contextual information, and gating mechanisms, which determine the influence of this information.

English Nominal Extraction and Coreference Resolution. Our English nominal extraction and coreference resolution were trained from ACE and ERE corpora, using embedding and distance features.

Cross-lingual Joint Entity and Word Embedding. Traditional methods of representing entity mentions consider each name mention as a common phrase, and use the combination of word embeddings of first name token and last name token. There are two major problems on these methods: (1) many names are out-of-vocabulary and they don't appear in the training data; (2) phrase embedding cannot disambiguate entities. We develop a novel cross-lingual joint entity and word method that not only can capture each entity mention as a single unit and perform disambiguation, but also allow all languages to share one common space. For each sentence in an IL Wikipedia, we replace each IL mention with the title of the English entity it's linked to, and then use this code-switch data to learn joint entity and word embeddings for IL; For each English Wikipedia sentence, we replace each mention with the title of the English entity it's linked to and construct the semantic space for English. In other words, the shared linked English entities are used as anchors to align two spaces. We gradually rotate the IL space so it can be aligned with the English space by learning a mapping function (rotation matrix).

Name Translation Mining. We mined IL-English name translation pairs from various approaches: (1) Cross-lingual Wikipedia titles; (2) Cross-lingual Geoname titles; (3) We collected incident-related comparable documents from Set 0, Set 1 and Set S. Then we measure the similarity between every pair of IL name and English name using the cross-lingual joint entity and word embedding, and discover name translation pairs. These mined name translation pairs are used in cross-lingual entity linking, Automatic Speech Recognition and Machine Translation.

Cross-lingual Entity Linking. After we translate each each IL name mention into English, we apply an unsupervised collective inference approach to link each translated mention to the target KB. The unique challenge in the LORELEI setting is that the target KB is very scarce, without rich linked structures, text descriptions or properties as in traditional KBs such as Wikipedia. We associate mentions with entities in the target KB in a collective manner, based on salience, similarity and coherence measures [7]. We calculated topic-sensitive PageRank scores for 500k overlapping entities between GeoNames and Wikipedia as their salience scores. Using the scores of overlapping entities, we calculated the average score of each geographical type, such as city, village, and lake, and thus estimated the salience scores of out-of-DBpedia entities using their type scores. We use the cross-lingual joint entity and word embedding as an additional similarity measure. Then we construct a knowledge networks from source language texts, where each node represents a name mention, and each link represents a sentence-level co-occurrence relation. If two mentions co-occur in the same sentence, we prefer their entity candidates in the KB to share administrative code and type, or close in terms of latitude and longitude values.

Cross-lingual NIL Clustering.

For NIL mentions we created initial clusters based on exact string matching on mention surface forms. Then we applied multiple steps to cluster mentions: (1) We developed a normalizer to normalize surface forms by removing name designators

and stop words and stemming; (2) We clustered mentions with similar NYSIIS representation (similar to Soundex) longer than four letters, after removing double consonants and vowels; (3) We clustered two mentions if the edit distance between their normalized surface forms is equal to or smaller than a threshold; (4) We clustered two mentions if their distance in the joint entity and word embedding space is shorter than a threshold; (5) Finally we merged two clusters if they include mentions sharing the same English translation.

C. Critical Additional Features and Tools Used

We incorporate both character embedding and contextualized word embedding [8] as features for name tagging. We used the results from universal romanization tool [9] for Chinese room annotation.

D. Other Data Used

We used a multi-lingual Wikipedia dump, and massively multi-lingual Panlex which were collected before the incident dates.

E. Significant Pre/Post-Processing

We used UPenn morphology analyzer [10] to segment Kinyarwanda words for Chinese room annotation. We added a post-processing step to extend the name boundaries to include GPE designators. We developed another post-processing step to process names in @ mentions and hashtags in tweets, by automatically parsing each mention into multiple tokens, and running English EDL to candidate names.

F. Native Informant Use

We run our IL name taggers on Set 0 and Set 1 and ask the NIs to translate frequent names which are not in our name translation gazetteers into English.

G. Remaining Challenges

We were not able to conduct more detailed error analysis because there is no ground truth or score feedback. The following are some challenges we have identified during analyzing Set 0, Set 1 and Set S name tagging errors. Many names are specific to incident-related regions and topics. It's challenging to acquire enough training data to cover these names and their contexts. These names are even out-of-vocabulary for state-of-the-art English name taggers. In the following sentence "*The final Perahera of the **Ruhunu Kataragama Maha Devalaya** will be held today.*", only after we see the image of **Ruhunu Kataragama Maha Devalaya** we can infer it's a temple and label it as a location. In "*In the communiqué the education ministry has cited as a cases in point several instances like the application by a doctor transferred to **Bemmulla** in **Gampaha** for admission of his child to the **Colombo D . S . Senanayake Vidyalaya***", **D . S . Senanayake Vidyalaya** looks like a person name based on its surface form, but it's a college. Similarly, without web search it's hard to know **Kerala Ganja Cannabis** is a drug in the following sentence "*The navy media unit stated that they suspect that the Kerala Ganja Cannabis was brought from India via the mainland*"; and **IOC** refers to **Indian Oil Corporation** in "*IOC's fuel prices will again rise again in the light of the increase in fuel prices in Ceylon Petroleum Corporation*".

To alleviate this domain knowledge bottleneck, an intelligent name tagger will need to be self-localized rapidly. The system should perform *automatic googling* beyond local context, by automatically linking each mention to a huge web-scale corpus and analyzing all related documents for knowledge discovery and embedding learning. For next year's evaluation we suggest NIST and LDC to provide as much IL and English monolingual data as possible. It would be even better if these monolingual corpora are in multi-media form.

III. Machine Translation

Our MT team consisted of Ronald Cardenas, Thamme Gowda, Ulf Hermjakob, Nima Pourdamghani, Michael Pust, Jibiao Shen, Tian Xie, Kenton Murray, Toan Nguyen, Heng Ji, David Chiang, and Jonathan May.

A. Core Algorithmic Approach

General components.

MT Systems. Our primary submissions were a combination of several MT systems within in the ELISA project. These were:

- A syntax-based MT system (SBMT) built at ISI.
- A hierarchical phrase-based system (Hiero) built at Notre Dame.
- A phrase-based system (Moses) built at Notre Dame.
- A recurrent neural system built at Notre Dame.
- Two convolutional neural systems based on Transformer (one at ISI, one at UW).

These MT systems were trained on parallel data provided by NIST. See section III-B for details. Translation models and language models used mixed-case data. Unlike in previous years, we did not use stemmed corpora for word alignment of the non-neural systems as the training corpora were significantly larger than had been seen before. Below we describe details of each of these systems.

ISI Syntax This is a string-to-tree statistical MT system based on [11], [12]. We used two word aligners (GIZA and Berkeley). We tuned with MIRA. We used a gigaword 5gram language model and an additional LM built from Leidos reliefweb.

ND Hiero This system was a hierarchical phrase-based system, trained on v1 of the data. Preprocessing, word alignments, and language models were the same as described above for Moses. We extracted hierarchical rules from all parts of the corpora likely to contain parallel sentences, and phrases from translation lexicons. We trained feature weights discriminatively using MIRA with feature scaling somewhat similar to RMSprop.

ND Moses Our Moses system relied on data preprocessed with Morfessor Flatcat. Our alignments were generated with

GIZA++ (both intersection and grow-diag-final-and) as well as the Berkeley aligner. All three were combined in order to generate the phrase table. We used three n-gram language models. These were built from the bitext, Gigaword, and Leidos using KenLM. Our model was tuned using MERT.

ND Recurrent The sequence-to-sequence system use the standard LSTM encoder-decoder with attention. We applied two enhancements, a normalization technique that fixes the norms of all word embeddings to some value, and a lexical module that predicts the target word based on only the source words. All systems were trained for 40 epochs with a batch size of 32, and a vocab size of 12119 word pieces. We used the full training data with lexicons.

ISI Transformer The transformer system [13] used direct tensorflow transformer code from Google. It was run with default hyperparameters except for a batch size of 4096, vocab (number of word pieces) of 32,768, and all training data (but no lexicons). It ran for 64,000 steps (minibatches). We also used this setup to build backtranslation models and generate backtranslation data from the Leidos Reliefweb corpus (but did not use this data in CP1). We noticed that Transformer appears to behave worse when the orthographies are not Latin. We have not yet determined why this is but suspect the word piece segmentation algorithm may be tuned for Latin characters. To get around this for Sinhala we used Uroman [14].

UW Transformer The UW Transformer NMT system built on the attention-only neural machine translation model of [13]. The system employs the subword modeling approach of the Transformer system. However, prior experiments on Tamil had indicated issues with non-Latin scripts, and thus we applied uroman [14] to romanize the Sinhala script prior to processing. We also trained the model on lowercase text, but built a corresponding recaser, paired with a detokenizer, to create final mixed case output. Based on prior tuning experiments on dryrun languages, our model used a batch size of 1024, a vocabulary size of 8192 word parts across source and target languages, learning rate of 0.1, two encoding and two decoding layers, and a hidden layer size of 512. The model was trained for 250,000 steps for each IL in CP1, and a number of steps determined by performance on the NISET data in CP2 (113,000 steps for Kinyarwanda ; 250,000 steps for Sinhala). For CP2, the parallel training data was augmented with the lexicon and upsampled gazetteer data.

B. Data

We trained MT systems on the following parallel data resources:

Language Resources provided by UW The LanguageNet massively multilingual lexical resource was used to augment the LDC-provided lexical resources for Kinyarwanda and Sinhala . LanguageNet "masterlexicons" have been compiled for the duration of the LORELEI project using publicly available Internet and print resources such as Wiktionary, PanLex, online dictionaries and language learning sites [15]. These include, minimally, a translation pair between a source and target language and the source of the translation pair. The target language is English in most cases; however, where translations into other languages were readily available from a source, these were also gathered, with each language pair having its own masterlexicon. The masterlexicons use the information exactly as given in the source material. Where the source material included other information, such as part of speech, transliteration, pronunciation, dialect or domain, these are also included. For each language pair, the lexicons gathered from different sources are merged into a single masterlexicon. Entries including the same word and translation are merged into a single entry, with all sources attributed. These entries were incorporated into our released 'lexicon' versions.

TABLE II
Parallel data versions. Words shown are English words in the parallel training set corresponding to the source language indicated.

Version	description	Kinyarwanda	Sinhala
1	initial auto-extract	4,151,376	6,438,717
2	tokenization fix	n/a	6,438,717
3	Gargantua re-alignment	4,034,444	6,165,523
4	duplicate segment elimination	3,765,588	5,865,482
5	comparable segment selection	3,021,096	3,095,721
6	cleaned monolingual before Gargantua	3,765,310	6,068,514

Bilingual dictionaries. We used the LDC-provided dictionary, and we pulled other entries from pre-collected massively multilingual resources. We cleaned dictionary entries (deleting infinitive "to" on the English side, etc). We received lexicons from UW, per the description above. We numbered our dictionary releases (v1, v2, v3, …v6). The dictionary sizes are shown in Table III. The initial version of dictionaries solely consisted LDC provided lexicons, and the subsequent releases were made by augmenting it with other entries. The statistics of final release of dictionary is given in Table IV.

Parallel Corpora. We processed the provided parallel data into sets called 'train' (for MT rule acquisition), 'dev' (for MT tuning), 'syscomb' (for multi-system combination), and 'test' (held out). As in previous years, we used a set of incident-relevant English keywords to choose the documents that went into test, syscomb, and dev, in that priority order. Previous years' LoreHLT evaluations did not have any incident data so this approach formerly did not do much, but in this year there were substantial news articles in Sinhala and even in the bible-inspired text that made up the bulk of Kinyarwanda so we were more confident that the test sets would be indicative of evaluation performance.

We also numbered our parallel data releases (v1, v2, v3, …). Table II denotes the various versions of our data, brought about by different cleaning approaches employed after previous versions were shown to yield suboptimal results. Below we describe the various versions of our data. All changes were applicable to both Kinyarwanda and Sinhala unless otherwise noted.

1) This was the initial version auto-extracted from parallel data. It was found to have a great many sentence pairs that were not translations of each other. The default cleaning filtered segments with length ratios and length deltas more than two standard deviations outside the mean, which is a conservative approach that assumes most segments are well-aligned. However, in the case of Kinyarwanda and Sinhala data the assumption was invalid and consequently little data was actually filtered. For example, in Kinyarwanda the automatic processing required ratios to be in excess of 8.2 and deltas to be in excess of 21.7; only 1% of lines met both criteria.
2) Sinhala Version 1 had faulty tokenization, due to a misunderstanding about the proper order of operations. This was a technical fix.
3) Starting in CP2, in Kinyarwanda and Sinhala , we used the Gargantua sentence alignment tool [16] to align data. At this point we re-split dev/test/syscomb data sets so Versions 1 and 2 are not compatible with this version and beyond. Thereafter we kept the same splits (but sometimes threw away or realigned segments) so subsequent versions are compatible.
4) We removed duplicate segments. The watchtower data had high incidents of duplication, which led to overconfidence in Kinyarwanda results (Sinhala test/dev/syscomb sets were generally news-only due to our incident-relevant set selection process). Any segment pair that appeared previously in data was eliminated (using the data set order train-dev-syscomb-test).
5) We decided to treat the parallel data as comparable and threw out the initial alignments. We developed a toolkit called ReAligner[1] which used a combination of rules and scoring functions to produce new alignments. The tool was scoped to align segments within documents. It performed so by scoring and ranking every combination of sentences within a given pair of source and target documents. We applied rules such as length ratio, punctuations count ratio, presence or absence of numerals, and URLs on both sides to reduce the exhaustive search of the re-aligner. Details of scoring functions and algorithms for this approach are below. We also filtered entire documents that were not parallel; this was particularly a problem for news data in Sinhala . We used NI time to inspect document headlines and asked the NI to say whether or not a headline was a reasonable translation, to filter out for document mismatch.
6) We noticed that many of the alignments in the LDC pack were seemingly inaccurate. For instance, we observed segments having numerals on the source side aligned to segments without numerals on the target side (and vice versa). Length ratios between source and target segments were suspiciously abnormal.In an analysis during which we detected unalignable URLs (URLs for which we cannot find the same URL on the other side) and tokens such as 'img' and 'MP3' on both source and translation sides and filtered segments containing those, we observed 11605 segments reduced from Kinyarwanda data and 16656 segments reduced from Sinhala data. We then ran Gargantua again, as in version 3.

[1]https://github.com/thammegowda/realigner

Re-aligner details For the re-aligner, we experimented with two approaches to score the sentence pairs: (i) MCSS Similarity score and (ii) Translation Table (T-Table) score.

MCSS Similarity score Multilingual Common Semantic Space (MCSS) is a system based on neural word embedding that projects embeddings from one language to the other. The word vectors were trained on a large monolingual dataset of English, Kinyarwanda , and Sinhala , followed by projecting Kinyarwanda and Sinhala word vectors to English embedding space. Intuitively, MCSS scoring function mapped source and target words into the same vector space, and computed sentence vector by aggregating the word vectors in the sequence. The MCSS similarity score was the cosine similarity between source and target sentence vectors.

Translation-Table (T-Table) score For a source sequence $s = s_1 s_2 s_3 ... s_m$ and target sequence $t = t_1 t_2 t_3 ... t_n$, the $score_{ttab}$ was computed as follows: The translation probability of lexicons was estimated using GIZA++ aligner. The dataset excluded parallel sentences from the LDC IL package, instead used the dictionaries and previously mined parallel sentences from the web by our massively multi-lingual web crawler. We generated T-table from both the sides, i.e. forward T-Table $\overrightarrow{T} : P(t_j|s_i)$, and inverse T-Table $\overleftarrow{T} : P(s_i|t_j)$. We ignored the casing of text by converting all the texts to lower case.

$$score_{ttab}(s,t) = \frac{1}{2}\big[\frac{1}{m}\sum_{i}^{m} score_{tok}(s_i, \overrightarrow{T}, t) + \frac{1}{n}\sum_{i}^{n} score_{tok}(t_i, \overleftarrow{T}, s)\big]$$

The $score_{tok}(w, T, c)$ was computed as:

$$\begin{cases} 0 & \text{if } w \notin T \text{ and } w \notin c \text{ i.e. OOV} \\ 1 & \text{if } w \notin T \text{ and } w \in c \text{ i.e. Copied} \\ \sum_{w' \in c} P_T(w'|w) & \text{otherwise} \end{cases}$$

Evaluating the re-aligner We also set up a test to evaluate the performance of our alignment functions. The test for alignment quality used 1000 sentence pairs which are properly aligned as positive samples, and randomly generated negative samples for each of those 1000 source sentences. We experimented with various sizes of negative sampling sizes; the easiest test had 20 negatives, and the hardest test had 200 negatives for each source sentence. For MCSS, the sentence alignment error was to be 13% and 35% with 20 and 200 negative samples. The T-Table scoring function had the sentence alignment error of 10% and 25% respectively for 20 and 200 negative samples. The internal parallel data release version v5 included T-Table based aligner. We set a threshold score of 0.2, based on our manual inspection

TABLE III
Dictionary versions and number of entries

Version	Kinyarwanda	Sinhala
1	48,333	227,668
2	76,679	249,833
3	90,379	Not released
4	89,345	240,526
5	90,228	241,476
6	90,285	241,797

TABLE IV
Statistics of final version (v6) of Kinyarwanda and Sinhala dictionaries

Source	Kinyarwanda Count	Sinhala Count
LDC IL pack	48,333	227,668
ISI-DICT	19,290	5,096
RPI-GAZv2	3,367	2,985
UW-Panlex	2,177	3,983
UW-others	292	1,317
NI-Phrases	360	427
UW-NI	57	321
RPI-Kinyarwanda.net	13,700	N/A
UW-BabelNet	2,709	N/A
Total	90,285	241,797

on bad alignments, to remove the misaligned data. As a result, our re-aligner found no valid alignments for $55,079$ out of $298,660$ segments in Kinyarwanda and $190,826$ out of $415,041$ segments in Sinhala datasets.

C. Other Elements

Morphology. The syntax MT used the *Morfessor* system for unsupervised splitting of words, as in previous years it was additionally exposed to full-word analyses in a source-language lattice. Penn's unsupervised morphology was not found to be as helpful in preliminary internal investigations. Most of the neural systems used byte-pair encoding as a sort of rough morphological splitting; we noticed that the BPE used by Transformer seemed to have some problems with the non-latin orthography of Sinhala so we applied Uroman pre-romanization to avoid these issues.

System combination. We used Kenneth Heafield's Multi-Engine Machine Translation (MEMT) software [17] to combine individual MT systems. The software constructs lattices from sets of translations by heuristically aligning words, then tunes weights for a set of language model and per-system features to optimize Bleu, using the MERT algorithm [18]. We tuned system combination using the 'syscomb' set and chose systems with high individual Bleu scores. We show 'syscomb' as a submission in all check points and languages; for these systems we also, show, separately, a table of component systems that went into the combination.

Handling of Sinhalese complex numbers

We extended a special lexicon-and-rule-based translation system to translate complex numbers from Sinhalese, which uses a mix of Western and Indian style number systems. Example translations (Sinhalese source shown in uromanized form):

- dekootti asuupan laksayaka ("two-crore eighty-five lakh-*suffix*") $\Rightarrow$ 28.5 million
- eklaksa hatalis hayadahas tunsiya tis hayak $\Rightarrow$ 146,336
- kootti 1574 yi dasama 5 yi 2 $\Rightarrow$ 15.7452 billion
- biliyana 15 $\Rightarrow$ 15 billion
- 3,45,67,890 $\Rightarrow$ 34,567,890

The output of this special system competes against other MT modules based on automatically tuned weights. Unfortunately, it did not appear to give a Bleu gain. We were unable to take time to analyze why this was so.

Document Selection: We devised several approaches to selecting documents for Chinese Room and/or Native Informant annotation. One approach is a semi-supervised variant of the CoReX algorithm. The other is based on pairwise mutual information and keyword expansion.

Chinese Room: We previously developed a interface (the "Chinese Room") [19] that allows monolingual English speakers to translate sentences from an arbitrary, unknown language, given a dictionary and a small parallel text. It makes these resources available in an intuitive way. We had previously hired and trained three USC Masters students (all of whom have several Indian language and English fluencies but none of whom speak the ILs). We also received help from a graduate student at RPI. All in all, the CR annotators performed slower than anticipated during the evaluation, yielding a total of 311 usable words of Kinyarwanda and 623 words of Sinhala , available after our NI sessions. We used this data as silver sets for determining which systems to submit, out of fear of overfitting on our native informant sets.

D. Use of Native Informant

Native Informant: The ELISA team received two units of NI time for this evaluation. The UW team received one unit. The general approach to use of NI was as follows:

- ELISA unit 1 was used in CP1 and CP2 to collect translations of English incident terms into the ILs.
- ELISA unit 2 was used in CP2 to collect whole-sentence translations of IL situational documents, both translating from scratch and from Chinese Room annotations.
- UW unit was used for...

Below we describe some details of each unit. All collection was used by all MT and SF subteams.

ELISA unit 1 For all sessions, we asked the NI to translate a list of English terms into their native language (Kinyarwanda or Sinhala). If a term could not be translated, the NI provided the best available translation and a comment of why the term could not be translated. The term list was selected from Leidos and OSC corpora, using a combination of class-relevance, document frequency and manual filtering. In total we had 10 NI sessions of 1 hour each (5 per IL). We worked with 4 informants: NI2 and NI4 for Kinyarwanda , and NI5 and NI6 for Sinhala . By checkpoint 1, NIs had translated 254 terms: 200 for Kinyarwanda and 54 for Sinhala ; by checkpoint 2,

TABLE V
Number of translations obtained per session date.

	Kinyarwanda	Sinhala
6/2/18	105	36
6/3/2018 - preCP1	95	54
6/3/2018 - postCP1	-	115
6/4/18	-	-
6/5/18	97	122
6/6/18	201	101
6/7/18	-	-
6/8/18	-	-
6/9/18	189	-
Total	687	428

they had translated 881 terms: 687 for Kinyarwanda and 428 for Sinhala . The breakdown of number of translations per session is presented in table V. Overall the NI sessions went without any delays or incidents, all NIs were professional and ready to do their jobs. The only exception was our last meeting on Monday 9th, where NI2 was unable to attend the call due to personal circumstances. Appen re-assigned us to a new time slot with NI4 on the same day without any further complications.

ELISA unit 2 NI3 (Kinyarwanda) did 4 hours, translating documents regarding floods, unrest, and drought, amounting to 52 sentences. NI4 (Kinyarwanda) did headline comparison and verified all JW articles in syscomb, dev, and test were valid. NI5 (Sinhala) 728 words in 2 hours. NI6 did 1324 words in 2 hours. The last hour of this unit for Sinhala ws used to discover that about half our news-oriented Sinhala articles in test were probably not real translations and 2/3 of the inspected syscomb sentences were similarly not translations of each other.

UW unit The UW/UIUC team was allocated one unit of NI time. The use of this time largely aimed to enhance Named Entity handling, especially geo-political and location entities, in support of lexical resource enhancement and language modeling for speech recognition. Using the (assumed) parallel English-IL training data, we applied an off-the-shelf English Named Entity Recognition system [20] to identify English-side entities as candidates for translation. The task was revised based on experience and observations about the set0 parallel training data.

- Kinyarwanda /Hr1/NI3: An alphabetized list of English PER/GPE/LOC entities was presented to the NI for translation to the IL. No additional context of occurrence was provided. The NI made rapid progress through the list of entities, but expressed concerns about orthographic variation in the English tokens.
- Sinhala /Hr1/NI7: An alphabetized list of English PER/GPE/LOC entities was presented to the NI for translation to the IL. No additional context of occurrence was provided. The NI made steady progress through the list of entities.

Beginning in Hr2, we made some revisions to the task protocol. Due to concerns about out-of-context translation and English-IL translation equivalence in the training sample, we began presenting the entities to be translated in-situ, in an example English sentence, paired with its presumed IL translation. Annotators were also given the option to indicate that no translation of the English NE term appeared in the presumed parallel text or that the sentences were not translations of each other. In addition, having observed the prevalence of personal names in the Watchtower text, we restricted the NE types for selection to GPE/LOC, which we hoped would be more generally relevant to the downstream EDL and SF tasks.

- Kinyarwanda /Hr2/NI2: In this iteration of the task, we asked the NI to also provide a translation of the English sentence if the candidate parallel sentence was found to not actually be a translation.
- Sinhala /Hr2/NI7: Here, in order to enrich the corpus of entity related utterances, we also asked the NI to provide a paraphrase of the NE-bearing sentences.

The translation/paraphrase variants of the task proved very time-consuming, with NIs only able to complete 7-10 sentences in the hour of time. NIs also commented that the longer sentences in these tasks were difficult to work with.

As a result, for the remainder of our NI time, we a) returned to focus on the NE translation task alone and b) restricted context sentences by length to 20 words or fewer. (Kinyarwanda /Hr3/NI2:Sinhala /Hr3/NI5: Kinyarwanda /Hr4/NI1:Sinhala /Hr4/NI6:Kinyarwanda /Hr5/NI1:Sinhala /Hr5/NI6)

The results of NI efforts augmented the IL gazetteers and informed the entity-targeted language modeling efforts.

Checkpoint 1.

Pre-and post-processing. It was extremely handy to have the *uroman* tool prepared in advance, so that we could view and process Sinhala in Latin script. Kinyarwanda is already in Latin script.

Noisy Data While we had an ample amount of parallel data in comparison to previous years, as noted above, it was rather noisily aligned by comparison. This was observed when inspecting the parallel data and when initial decodes consistently produced artifacts such as URLs that did not occur in source sentences. We started aggressively cleaning training data more than the default cleaning procedures built into our data processing pipeline at around the 14 hour mark of CP1 but did not make it into systems submitted in this checkpoint; systems from this checkpoint all use rather misaligned data.

Reduced training. Unlike in previous years, the parallel data provided to us was ample (3–6m words). In fact, we were concerned that we wouldn't be able to build MT systems in enough time for SF to make the checkpoint. We thus reduced our syntax training set. We automatically grew our training set by starting with a seed subset of training and then automatically adding data based on improving coverage relative to the evaluation set. This is all done without human intervention or manual inspection and is equivalent to targeting 'important words and phrases' as noted in clarification emails.

Backtranslation We've found it beneficial for certain types of MT (e.g. Transformer) to augment or domain-shift low-resource training data with English that has been back-translated into the source language using an earlier version

TABLE VI
Kinyarwanda CP1 system submissions

	dev	test	syscomb
t2t-v1	13.0	15.7	19.0
nd-hiero-s2i7	15.3	9.0	17.2
isi-sbmt	11.2	14.6	14.3
nd-nmt_norm_bridge	13.5	15.2	17.6
nd-moses	6.8	7.7	8.2
t2tbt-v1	12.1	16.5	17.9
uw-tfm	6.6	10.4	9.8
combo	16.8	18.1	20.7

TABLE VII
Sinhala CP1 system submissions

	dev	test	syscomb
t2t-uroman-v1	4.5	3.3	3.4
isi-sbmt	3.2	2.8	2.5
nd-hiero-s1i7	4.3	3.4	2.9
t2t-v1	3.1	2.9	2.5
nd-nmt_lex	5.1	3.6	3.7
uw-tfm	2.7	2.1	1.7
nd-moses	4.4	3.1	3.1
combo	5.5	4.8	5.1

of a symmetric mt system. We back-translated 6.6m words of Leidos Reliefweb data into Kinyarwanda and Sinhala and used this data, together with the provided parallel data, to build Transformer engines.

Re-scoring. Our neural MT (NMT) systems did reasonably well in standalone mode, so we did not make an attempt to use them to re-score SBMT n-best lists this year, since that procedure is somewhat more time consuming.

Submitted results' proxy scores on internal sets are shown in Tables VII and VI. Additionally, the components that went into the systems are shown in Tables X and XI. All systems are constrained.

Lessons Learned:

- Having a relatively large data set is not a panacea if it is extremely noisy
- The old trope about fighting the last war is true and seems almost inevitable; handling unknown unknowns a priori currently is AI-complete and human intervention is inevitable.

TABLE VIII
Kinyarwanda CP2 system submissions

	dev	test	syscomb	niset	crset
t2t-lgb-v4	28.4	30.1	28.4	7.1	13.1
nd-hiero-s7	2.6	3.0	2.7	0.3	12.3
isi-sbmt-v5-small-mixed	23.6	22.5	20.7	5.3	12.0
t2t-lgbxv-v5	27.6	30.1	28.2	5.9	12.6
uw-tfm-6	8.5	8.8	7.9	2.7	4.5
combo3		31.5	30.1	7.2	13.8

TABLE IX
Sinhala CP2 system submissions

	dev	test	syscomb	niset	crset
t2t-lgb-v5	9.4	9.9	8.6	9.8	12.3
isi-sbmt-v4-mixed		9.2	8.4	10.8	11.8
nd-hiero-s7		6.1	4.8	6.8	6.8
t2t-xuroman-lgb-v5	8.6	9.0	8.1	10.1	8.5
nd-nmt_lex	9.3	9.6	8.9	10.1	8.9
combo3		12.2	10.9	11.9	
nd-moses-v6	6.5	6.1	6.2	7.0	8.9
uw-tfm-6	9.1	7.2	7.4	7.1	7.1

TABLE X
Components of Sinhala CP1 system combination submission (v2 data used)

	dev	test	syscomb
isi-sbmt	3.2	2.8	2.4
nd-hiero-s1i7	4.3	3.4	2.9
nd-nmt_lex	5.1	3.6	3.6
t2t-uroman-v1	4.5	3.2	3.4
t2t-v1	3.1	2.9	2.5
System-combination	5.5	4.8	5.1

TABLE XI
Components of Kinyarwanda CP1 system combination submission (v2 data used)

	dev	test	syscomb
isi-sbmt	11.1	14.5	14.2
nd-hiero-s2i7	15.3	9.0	17.2
t2tbt-v1	12.0	16.2	17.7
t2t-v1	13.0	15.5	18.8
System-combination	16.8	18.1	20.7

TABLE XII
System combination (combo3) submission for Kinyarwanda CP2 (v1 data used)

Kinyarwanda	test	syscomb	niset	crset1
isi-sbmt-v5-small-mixed	22.4	20.6	5.3	11.9
nd-hiero-s7	32.8	30.1	2.9	12.3
t2t-lgb-v4	29.7	28.2	7.1	12.9
t2t-lgbxv-v5	29.8	28.1	5.6	12.6
system combination	31.5	30.1	7.2	13.8

TABLE XIII
System combination (combo3) submission for Sinhala CP2

	test	syscomb	niset
isi-sbmt-v4-mixed	9.2	8.3	10.8
nd-hiero-s7	6.1	4.8	6.9
nd-moses-v6	6.0	6.2	7.0
nd-nmt_lex	9.4	9.0	10.1
t2t-lgb-v5	9.8	8.6	9.8
t2t-xuroman-lgb-v5	9.0	7.9	10.1
system combination	12.2	10.9	11.9

Checkpoint 2.

Parallel Data and Dictionaries. Cleaning parallel data became of utmost importance. We built versions 3 and beyond, as noted above.

In-domain data. We developed small domain parallel sets using the Native Informants (called "niset"), partially by working inside the Chinese Room with them. the Sinhala NIs in particular were surprisingly fast and in 4 hours of direct translation (Chinese Room sets were not yet available) we were able to produce almost 2200 words of translations of news articles. We also built sets using the Chinese Room that were not validate by NIs; these are called "crset".

Tweet Set:

We had no parallel twitter data and did not select any twitter data for NIs to translate. However, in order to get a sense of any systematic problems with our translations of tweet data, we selected, with RPI's automatic and manual help, a small set of Kinyarwanda and Sinhala tweets that we looked at various system translations of.

Transformations to Transformer We were able to make good use of the Transformer (also known as and interchangeably referred to as 'tensor-to-tensor,' 'tensor2tensor,' or 't2t') neural machine translation system [13], however there were some task-specific data properties that prevented us from using t2t out of the box. We made the following modifications:

- **Social Media Entities**: tensor2tensor uses BPE [21] to preserve some in-words information. But this will mess up entities found in Twitter and other kinds of short social media communication that should not be translated such as hashtags, handles, urls, and emojis. One solution could be add some pseudo copy-paste pairs to the training set, but since training a t2t takes a long time as the data size grows, we chose to pre-process the source, extracting those tokens we do not want to translate, then put them back after translation. This was a quick fix and altered some word order but seemed like the best decision given our time constraints.
- **Applying back-translation** It has been known that low-resource Transformer systems can benefit from incorporating additional target language data paired with an artificial source translation. Such a translation is typically obtained by building a target-to-source MT system and 'back-translating' the target data. Using in-domain data where available is ideal; we used the incident-rich Leidos Reliefweb corpus. We used tensor2tensor's transformer_base settings and trained the model for 128,000 steps on our training releases. We retrained whenever new training data or new vocabularies were available (see elsewhere for a discussion of this data construction).
- **Romanization**: When translating a non-latin language (e.g. Sinhala), we observed improvements by first turning the source language characters into latin letters which could represent their pronunciation, using uroman[14].

Below are some specific Transformer systems we built during this exercise:

- System t2t-lgb-v4 for Kinyarwanda : This system was trained on training data v4 + lexicon v4 + glosbe data + back translation v3. We set the training steps to 256,000, batch size to 4,096 and used the same transformer_base settings to train 8 models, and picked the best (based on our gold label test sets) to produce this part of outputs.
- System t2t-lgbxv-v5 for Kinyarwanda : this system was trained on training data v5 + lexicon v5 + glosbe data + back translation v4. We set this model's BPE vocabulary size to 65,536 (twice as the default) as we observed this would help Kinyarwanda translation. The reason could be Kinyarwanda had a larger vocabulary size. We trained it for 256,000 steps with 4,096 batch size.
- System t2t-lgb-v5 for Sinhala : this system was trained on training data v5 + lexicon v5 + glosbe data + back translation v4. We set the training steps at 256,000 and batch size at 4,096 to train the model and produced outputs for this part.
- System t2t-xuroman-lgb-v5 for Sinhala : this system is more complicated than above ones. It used romanization to preprocess the data. In combining romanization with back translation we have 2 options: 1) use romanized data to build reverse model and produce romanized translation; 2) use original data to build reverse model and romanized its translation. By doing option 1 we found that romanized model was usually better without back translation data but it became not as competitive after adding back translation produced by romanized reverse model. We suspected that the perplexity introduced by romanization was amplified by the not-so-good back translation process. Then came this system's idea. It was trained for 256,000 steps with batch size 4,096 on training data v5, lexicon v5, glosbe data, and back translation v4, all of which were romanized.
- Systems we tried but didn't use: systems based on v3 and v6 training data (not competitive), systems with different dropout rates (no obvious improvements), systems with larger hidden sizes (took too long to train and no obvious improvements), and systems with smaller BPE vocabulary sizes (not good).

Submitted results' proxy scores on internal sets are shown in Tables IX and VIII. Additionally, the components that went into the systems are shown in Tables XIII and XII. All systems are constrained.

Lessons Learned.

- It was important to distinguish parallel documents from non-parallel; we found that we couldn't completely trust the selection provided to us.

E. Critical Additional Features and Tools

As in previous years it was informative to look at output and compare output across different systems in regular group sessions. We looked at our automatically selected syscomb and test sets (subselected from the provided parallel data), translation of our NI-provided gold and CR-provided silver

sets, and even translation of tweets, for which we had no human translation of any kind. By doing this we learned:

- By virtue of many spontaneous unmotivated artifacts from all systems, that there were many misaligned sentences where multiple sentences that included much unrelated data was aligned to a single sentence and vice versa.
- By virtue of many translation system outputs agreeing with each other and strongly disagreeing with the reference, that our so-called 'parallel' data was comparable and not parallel.
- By simple observation, that part of one mt system had a bug and was copying its input without translating (this system was fixed)
- That simple copyable elements in tweets such as hashtags, user handles, urls, and emojis were not consistently being copied. We refined our postprocessing code developed in previous years to ensure these were all handled. Of particular interest was that uroman over-romanized and replaced, e.g., smiley face unicode characters with the romanization 'face'.

F. Other Data

We selected name pairs from our pre-collected, massively multilingual name pair list, derived from Wikipedia sources.

G. Filtering and re-alignment

see above sections.

H. Data Pre- and Post-Processing

See above sections.

I. Remaining Challenges

It is difficult to run in a truly automated fashion. The challenges vary considerably from year to year and human intervention is always needed. We would like to avoid this necessity.

IV. Situation Frames from Text

The primary team consisted of Nikolaos Malandrakis, Ruchir Travadi, Karan Singla, Victor Martinez, and Shrikanth Narayanan. However since the situation frame model used the name tagging and machine translation systems as modules, all members of the ELISA team have a contribution.

We submitted constrained and unconstrained runs of situation frame detection, including types, localization and status.

A. Core algorithmic approach

We implemented a variety of models targeting situation frames of different scopes, described below. The primary submissions were, for all checkpoints, combinations of one SF text and one SF speech model output, with "MLP-LSA", "CNN-GRU" and "MULTI" used for SF text. Our models are not multilingual: they can only process English and depend on the existence of machine translation and name tagging components, which they use as inputs. In all cases we used our team's translation and name tagging systems as inputs of the situation frame models. The models are top-down: they start by assigning types & status variables to documents and then attempt to localize these to the available locations, creating frames. Compared to the 2017 iteration of the task, this year's models are all multi-task, meaning they generate SF types jointly with SF status labels per type (last year we used a two-pass solution with a separate status model). We also switched to a supervised combination model and, of course, had to repeat the entire hill-climbing for hyper-parameter and data selection with nDCG as the primary performance metric. The models were trained on a combination of text and speech SF datasets, with hill-climbing used to decide which subsets of the data to use in each case.

Type & Status Detection Models

An overview of all models is shown in Fig. 1.

a) The CNN-GRU model: is a compositional CNN-GRU that accepts input documents as sequences of 1-hot vectors and uses a CNN to compose word embeddings into sentences and a single forward GRU to compose sentences into documents. It was pre-trained on the ReliefWeb and OSC corpora and the word embeddings were initialized using the, publicly available and general purpose, GloVe embeddings. Then the final layer was replaced and the entire network re-trained using a combination of SF speech and text data. The final layer is composed of 44 binary classifiers, corresponding to Type, Status, Resolution and Urgency with the latter three being produced separately for each type.

b) The MLP-LSA model: is a multi-layered perceptron applied to LSA document vectors. The LSA transformation was learned using the ReliefWeb & OSC corpora, which were also used to perform the first stage of training of the network. The second stage involves replacing the final layer of binary classifiers and re-training a combination of SF speech and text data.

c) The MULTI model: is the combination of the CNN-GRU and MLP-LSA models. The two constituent models are tied, by concatenating their first and last layers and then trained as a single network, using a combination of SF speech and text data.

Localization

The models described above are top-down: they consume the entire document and produce document-level labels. To localize, we use a simple solution of creating location-specific sub-documents and attempting to classify them using the same models. Given a detected LOC or GPE entity, we will collect all sentences/segments that contain said entity and form a

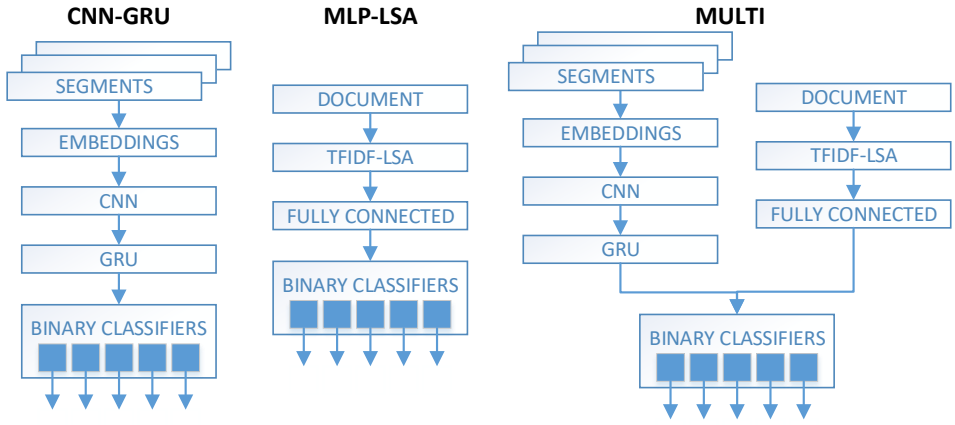

Fig. 1. Overview of the three SF Type+Status models

dummy "document" out of them. Then this dummy document will be passed through the same model and labels will be generated and then filtered by the complete document labels: a dummy document is not allowed to contain a type or status value that was not contained in the complete document. The final labels assigned to the dummy document corresponding to an entity mention are assigned to the entity mention itself. If no entity mention is connected to a type that was detected at the document level, then a non-localized frame is created for the specific type.

B. Critical data and Tools

The data used during development were:

- the publicly available GloVe word embeddings were used to initialize neural network embeddings
- the ReliefWeb and OSC corpora of disaster-related documents were used to train models
- the HA/DR lexicon was used for term and data selection
- an internal dataset of about 4000 annotated English tweets was used to train models
- the representative Mandarin, Uyghur, Oromo and English text SF datasets were used train and evaluate models.
- the transcribed and translated speech SF sets for Turkish, Uzbek, Mandarin, and Russian were used to train models.

The main tools and software packages used were:

- Python libraries: NLTK, gensim, Theano, Tensorflow, Keras, sklearn
- Matlab

C. Native informant use

As we were allotted two SF units and two MT units of NI time, and as much of the text SF collection activity is relevant for MT purposes as well, we prioritized speech collection over text and leveraged MT NI collection to aid text SF in CP1. Please see Section III-D for information about MT annotation used for text SF. Please see Section V-B3 for information on how the SF units were used.

D. The evaluation

Before the Evaluation

On the run up to the evaluation we identified the main challenges to this year's task. Below is a short list and how we tried to address them during the development phase.

1) *Only frames localized to KB IDs are taken into account.* We only generated localized frames and included a cross-reference check to validate the the entity we are localizing to is linked to an existing KB entry.
2) *Status variables are much more important. Since no credit is given unless all SF fields are correct.* Last year we used a 2-step solution, where we generated Type+Place frames, then assigned status to each as a separate step, that did not perform well, barely ever producing the minority labels. This year we integrated the status variable production into the SF Type model, so each for each Type we also generate the three status variables. To allow for the use of data with missing labels (almost all speech data are missing Urgency labels) we switched to a multi-task learning framework, where each Type and it's corresponding status variables are treated as separate tasks. This lead to much improved nDCG scores and also allowed us to address some of the following challenges.
3) *We have almost no speech data with Urgency annotations*. Since this is a multi-task setup, we used the text datasets to learn Urgency tagging for speech. We could not evaluate this approach.
4) *We have virtually no data with Urgency annotations for Issue Types.* Not much we could do. We wanted to produce the majority class label, but we do not know what that was. This was handled at the submission stage, by making submissions for both possible assumptions.
5) *The Urgency annotation protocol has changed significantly, potentially invalidating the little data we do have.* No choice. We had to assume the Urgency annotations would be compatible with previous years or we would

have no data to work on.

6) *The new scoring metric is impossible to tune for, since it has manually set hyper-parameters (the gain assignment thresholds) which we do not know how to account for and which can dramatically alter the scores. Also apparently the scorer was very buggy.* Again not many alternatives. We used some thresholds to develop against and hope that they were representative of the final thresholds.
7) *Due to incompatible localization annotations & lack of Urgency, no speech data could be scored with the new metric.* We used the 2017 metrics to evaluate SF speech. However the lack of compatible data also meant we could not evaluate any combination of speech & text data, so we had no way tuning the joint output.

The changes to the task lead to significant insurmountable challenges that we hope can be addressed for next year's task. As it stands we may be taking the low-resource premise of the task way too far: we need some compatible data in at least one language and we don't have that.

Checkpoint 1

At the beginning of checkpoint 1 we got access to the first batch of development data and the incident description document.

It was clear from the incident description that the issue SF types would be particularly significant for this scenario. This posed a serious problem for us: we had no training samples for issue frame urgency and this year's metric is very dependent on urgency in particular. We wanted to submit the majority class in all cases, but we did not know what that was. Our solution was to submit duplicates: submit the exact same SF output twice, once with all issue frames set as not urgent and once with all issue frames set as urgent.

This checkpoint proved very challenging due to the short time allowed. Development was frantic and any discovered issues or bugs had to be addressed very quickly - if they could be addressed at all. Due to the SF systems' position at the end of the overall ELISA pipeline, we could not really do much until very late in the evaluation period. We performed some sanity checks on the Set0 documents, but the only change made was that we dropped the hashtag splitter due to some bug that could not be addressed in the remaining time. This rush also had an effect on our input selection. Ideally we would use one EDL output and one MT output to produce SF, the best of each, but we had some concerns so used two MT outputs instead (separate submissions).

In total we submitted 20 runs, all constrained though only 10 will count as such. The main 18 submissions were generated by taking the Cartesian product of three sets:

1) SF text models: {CNN−GRU, MLP−LSA, MULTI}
2) SF speech models: {CNN−GRU, MLP−LSA, MULTI}
3) Issue Urgency: {urgent, not−urgent}

This lead to $3 \times 3 \times 2 = 18$ submissions. The last two submissions were created using a different MT from the first 18 and the "MULTI" models for both speech and text. An overview of our submissions can be seen in Table XIV.

Lessons Learned:

- 24 hours is a very short amount of time. This was particularly obvious at the end of the pipeline. The SF systems had to wait for inputs, each of which took time to produce and that wait took most of the first 24 hours.
- We really need compatible data so we can evaluate on the entire task instead of only part of it. We had to use all our submissions to account for things we had no data for, like issue urgency.
- It is unfortunate that the evaluation datasets focus on the more problematic areas of the task. Annotator agreement for issue types has been very poor, but the IL incidents focus on these types. Urgency agreement has been terribly poor & the annotation guidelines have changed, but the evaluation metric focuses on Urgency. This disconnect between resources and requirements should be addressed going forward.

Checkpoint 2

For SF text the extra time of between checkpoints was used to debug issues discovered during first checkpoint. We revised the hashtag parser, and used Set0 and 1 to validate. The results were satisfactory and the parser was used for all CP2 submissions. Other than that, no changes were made to the SF text models, so any improvements have to come from improved input (MT and EDL) quality.

The submission strategy was the same as CP1, with 18 submissions made using the Cartesian product of SF text and SF speech systems and 2 more submissions made using alternative MT inputs and the "MULTI" models.

Lessons Learned:

- No amount of debugging is enough. The SF system sits at the end of a very complex pipeline and any preceding component can introduce unforeseen issues. This was more of problem for CP1, but we could not really address any of the issues until CP2.

E. Remaining Challenges

- We are still very dependent on MT performance. We expected to have some MT-independent components for this evaluation, but they never reached the required performance. We will hopefully have them ready by next time.
- With increased data we saw improved performance from the more complicated networks. We expect that trend to extend into the future, as more data is released. Hopefully that will allow us to use more ambitious approaches.
- Perhaps in the future we can have data representative of all the tasks we are expected to perform. We have the knowledge and the infrastructure, but there is only so much we can get out of nothing.

V. Situation Frames from Speech

To produce situation frames from speech we followed a similar approach with the one described in the previous

TABLE XIV
ELISA IL9 and IL10 SF Submissions

Check Point	Condition	Submission (IL9/IL10)	Description
1 & 2	Constrained	200/210	(Text,Speech) = (CNN,MLP)
1 & 2	Constrained	201/211	(Text,Speech) = (CNN,MULTI)
1 & 2	Constrained	203/208	(Text,Speech) = (MLP,MLP)
1 & 2	Constrained	204/213	(Text,Speech) = (MLP,MULTI)
1 & 2	Constrained	206/215	(Text,Speech) = (MULTI,MLP)
1 & 2	Constrained	207/216	(Text,Speech) = (MULTI,MULTI)
1 & 2	Constrained	219/228	(Text,Speech) = (CNN,MULTI), Issue frames set Urgent
1 & 2	Constrained	225/234	(Text,Speech) = (MULTI,MULTI), Issue frames set Urgent
1 & 2	Constrained	254/256	(Text,Speech) = (MULTI,MULTI), Second choice MT
1 & 2	Constrained	255/257	(Text,Speech) = (MULTI,MULTI), Second choice MT, Issue frames set Urgent
1 & 2	Unconstrained	199/209	(Text,Speech) = (CNN,CNN)
1 & 2	Unconstrained	202/212	(Text,Speech) = (MLP,CNN)
1 & 2	Unconstrained	205/214	(Text,Speech) = (MULTI,CNN)
1 & 2	Unconstrained	218/227	(Text,Speech) = (CNN,MLP), Issue frames set Urgent
1 & 2	Unconstrained	217/226	(Text,Speech) = (CNN,CNN), Issue frames set Urgent
1 & 2	Unconstrained	220/229	(Text,Speech) = (MLP,CNN), Issue frames set Urgent
1 & 2	Unconstrained	221/230	(Text,Speech) = (MLP,MLP), Issue frames set Urgent
1 & 2	Unconstrained	222/231	(Text,Speech) = (MLP,MULTI), Issue frames set Urgent
1 & 2	Unconstrained	223/232	(Text,Speech) = (MULTI,CNN), Issue frames set Urgent
1 & 2	Unconstrained	224/233	(Text,Speech) = (MULTI,MLP), Issue frames set Urgent

sections for text documents. An overview of our system is presented in Fig. 2. The machine translation (MT) and name tagger (NT) components were presented in Sections III and II respectively. The automatic speech recognition (ASR) component is language specific and its output is passed to the MT component to be translated into English as well as the NT component to extract information regarding place mentions. Additionally, a relevance classifier can be optionally applied to the audio input stream, and gives information if an incident is present in the audio document. Application of the relevance classifier alters the training procedure as we will explain in the following subsections.

A. UIUC Automatic Speech Recognition (ASR)

To transcribe speech, UIUC's Mark Hasegawa-Johnson and Camille Goudeseune used their new high-speed speech recognizer ASR24 (https://github.com/uiuc-sst/asr24). Instead of taking a day or more to train an acoustic model from the speech data, it uses a pretrained model built by Krisztián Varga as an extension of the ASpIRE chain model, part of the standard ASR toolkit Kaldi. ASR24 combines this with a pronunciation dictionary and a language model, built from raw text and a table of grapheme-to-phoneme rules (G2P).

The word-trigram language model was built with standard SRILM tools. The raw text came from the ORIG_RAW_SOURCE elements in the IL xml files. It was cleaned up by various simple heuristics and by discarding text that used graphemes unknown to the G2P, such as words in completely different alphabets. Each IL's G2P had been built beforehand, by scraping Wikipedia (https://github.com/uiuc-sst/g2ps) and then converting IPA phones to the acoustic model's ASpIRE phones.

All processing was done on a single dedicated compute server. Combining the models into an ASR was the speed bottleneck, taking about 2 hours per IL. This was a surprising slowdown, perhaps because the training text was so large. Once combined, the ASR transcribed each IL's speech in just 30 minutes, using all 56 cores of the server. 2.5 hours after the start of the eval, the first Kinyarwanda transcription set was completed. 5 hours after the start of the eval, the first Sinhala transcription set was completed. 6 hours after the start of the eval, the second Sinhala transcription set was completed.

The second set's improvements were in its training data. (Because of user error, Kinyarwanda had *only* a second set; its first set's files were accidentally deleted.) The improvements were removing anything that looked like Bible verses, and adding phrases from Heng Ji's gazetteers.

For later checkpoints, UIUC will retranscribe the audio. Instead of ASR24's own quick-and-dirty LM, it will use a sophisticated LM built by UW's Gina-Anne Levow, improved gazetteers, a G2P that tolerates loanwords, and a systems redesign to allow for mixed-case words, which integrates more nicely with the mixed-case MT engines.

B. BUT Automatic Speech Recognition (ASR)

The BUT ASR system training was mainly based on exploiting the NI's. All experiments mentioned below are trained using KALDI toolkit [] and grapheme to phoneme converter (G2P) from [22]. We follow the direction of the previous evaluations, and make use of the advanced text-based system as described in the previous sections.

1) IL9—Kinyarwanda:

a) Data description: NI data collected (in both sessions) - 7 hours of recorded speech Text for language modeling - 0.4 million utterances and size vocabulary of 42k The vocabulary and LM text includes data from Gazetteers and web data

b) Input features: The speech signal was pre-processed using multilingual+music VAD (Voice Activity Detection) to

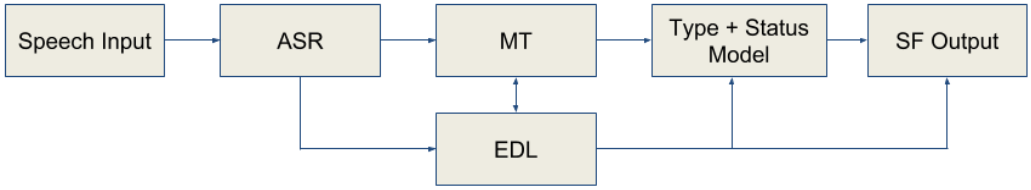

Fig. 2. Speech System Pipeline. Speech in the incident language goes through an Automatic Speech Recognition (ASR) component, whose output is utilized by the Machine Translation(MT) engine and the Name Tagger (NT). Once we have the translated output in English, as well as the place mentions, we identify the types of incidents occurring in the document and produce situation frames.

discard music and non-speech portions. Multilingual-RDT (MultRDT) features [23] and perceptual linear prediction (PLP) features were used for the experiments. The Multilingual RDT was trained initially with 17 languages from the BABEL corpus.

c) Acoustic model: Two speech recognition systems were built for the evaluation:

1) First system was a multilingual system built with Swahili and Zulu to output character sequence for the eval set. An unsupervised grapheme to phoneme converter (G2P) [22] was used for mapping multilingual output characters to characters in Kinyarwanda. The 7 hours of NI data is used for initializing the G2P system.
2) The second system was a basic DNN-HMM system built with Swahili and Zulu, and later ported to IL9. The alignments for the model were obtained using GMM-HMM model.

d) Language model: A tri-gram language model was built using the provided monolingual corpus and Gazetteers separately. These two models were later interpolated with equal weights to obtain the final model (based on the perplexity with the development set). This operation was performed to get an adapted language model towards the in-domain data. The lexicon contains 42k unique vocabulary words. Number of word tokens was chosen empirically to contain reasonable number of words and at the same time make the task computationally feasible. The words were chosen based on their occurances in the monolingual corpus.

2) IL10—Sinhalese:

a) Data description: Training data were provided solely by NI informants (see Sec. V-B3 below). The corpus contains 1389 utterances in 4.95 hours. Text for language modeling - 0.2 million utterances and size vocabulary of 42k The vocabulary and LM text includes data from Gazetteers and web data

b) Input features: The speech signal was pre-processed using multilingual+music VAD (Voice Activity Detection) to discard music and non-speech portions. Multilingual-RDT (MultRDT) features [23] and perceptual linear prediction (PLP) features were used for the experiments. The Multilingual RDT is trained intially with 17 languages from BABEL corpus.The major part of the training is similar to procedure in [24].

c) Acoustic model: Considering the limited amount of data for Sinhalese, we used 20 hours of Tamil data from BABEL and along with Sinhalese data for better generalization. The Tamil characters were transliterated from its own characters to Sinhalese characters and then used as initial model. This prior model was then ported to Sinhalese with data obtained from the NIs. Simple GMM-HMM was used for Sinhalese, unlike Kinyarwanda.

d) Language model: For IL10, we used the same procedure as for IL9, i.e. build the LM as an interpolation of two specific tri-gram LMs. The lexicon again contained 42k unique vocabulary words.

e) UW Entity-targeted Language Modeling: The Situation Frame task places significant emphasis on entity, especially geographic entity, recognition. Previous dryrun experiments had highlighted challenges in speech recognition for these classes of terms. As a result, we focused on development of language modeling approaches that would target and enhance speech recognition of named entity terms. All models employed the SRILM toolkit[25].

Five distinct strategies were investigated:

- Frequency boosting: This strategy increased the unigram frequency of the gazetteer terms by adding multiple instances of each IL gazetteer entry to the language model training corpus. This configuration was the primary ASR24 language model.
- Frequency boosting with entity-based data augmentation: In addition to the frequency boosting above, this strategy augments the language modeling training corpus with additional entity-bearing sentences. Candidate entity bearing sentences are identified as the translation parallel IL sentences corresponding to English language sentences in which entities were found by an off-the-shelf English named entity recognizer.[20] The additional sentences are created by probabilistically duplicating existing entity-bearing sentences and generating new entity bearing sentences by stochastically replacing entities in original sentences, found by gazetteer match, with alternate IL gazetteer entries. The rate of data augmentation was tuned on a pair of development sets, one targeting entity-dense sentences and another the overall corpus distribution.
- Supervised expansion of unsupervised class-based models: Class-based language models were created by unsupervised clustering over the language model training corpus, using a threaded implementation of Brown clustering [26], [27], [28]. Additional, previously unseen entries from IL gazetteers and GeoNames were added to the clus-

ter with the highest proportion of the corresponding class, with uniform frequency. The class-based model was then interpolated with a word-based tri-gram language model with modified Kneser-Ney discounting to create the final model.

- Supervised seeding of class-based models: Classes were initialized with entries from IL gazetteers and GeoNames, and Brown clustering was performed, building on that assignment. All gazetteer/GeoNames instances appearing in the training corpus were treated as tokens of that class; tokens not attested in the corpus were added with uniform frequency to the resulting cluster. The class-based model was then interpolated with a word-based tri-gram language model with modified Kneser-Ney discounting to create the final model.
- Supervised class creation: Classes were created and populated based only on IL gazetteer and GeoNames entries; all other tokens were treated as individual words. Tokens not attested in the corpus were added with uniform frequency to the corresponding cluster. The class-based model was then interpolated with a word-based tri-gram language model with modified Kneser-Ney discounting to create the final model.

For class-based models, configurations with the number of clusters ranging from 750 to 2000 were created. The configuration with the highest gazetteer term cluster purity was chosen, 750 clusters. However, based on perplexity over a development set drawn from Set1 data, the models were shown to be relatively insensitive to number of clusters, though the models which incorporated unsupervised clustering to build full class-based models outperformed those which only used supervised class creation.

Since the available training data was significantly larger than that in previous years and morphological complexity further contributed to a larger vocabulary size, we found that the resulting class-based models were also substantially larger, sometimes infeasibly so, especially with the conversions required for integration in the speech recognition pipeline. In some cases, this resulted in a prohibitive slowdown of the decoding process, even when trained on an entity-focused subset of the training corpus. The frequency boosting and entity-based data augmentation strategies were not subject to this issue.

For CP2, frequency boosting was applied to all language model training sets.

3) Native Informant for Speech: The strategy for us was to obtain as much training data for the acoustic models as possible. Following our previous strategy, we used the Native Informant (NI) for reading only. In the reading sessions, the NI's were asked to read sentences that were chosen from the Set0 text. The sentences were chosen based on the frequency of incident-related English-translated keywords, and were provide by RPI. The list of filtered sentences was then numbered and formatted into a googledoc (note that we had to discard sentences with any numbering or non-IL and unpronuncable text). The NI's were instructed to read the number in English and the sentence in their language. We used Audacity to capture the whole session, after which manual segmentation was performed based on the English numbers.

This way, we conducted 10 reading sessions per IL (2 in CP1 and 8 in CP2). The statistics are shown in Tab. XV.

TABLE XV
NI speech statistics. (clean speech in hours)

IL	CP1	CP2	Total
Kinyarwanda	1.49	5.51	7.00
Sinhala	1.10	3.85	4.95

C. Critical data and Tools

The data used during development were:

- the publicly available GloVe word embeddings were used to initialize neural network embeddings
- the representative Mandarin, Uyghur and English text SF datasets were used train and evaluate models.
- the ASR-transcribed and translated speech SF sets for Turkish, Uzbek, Mandarin, Amharic, Uyghur, Russian and IL6 were used to train models.
- BABEL corpus was used to obtain data for Tamil, Swahili and Zulu

The main tools and software packages used were:

- Python libraries: NLTK, gensim, Theano, Tensorflow, Keras, sklearn
- KALDI, G2P toolkit from BUT [22]
- Tamil to Sinhalese transliteration tool[2]
- R libraries: xgboost
- Matlab

D. Native informant use

See Section V-B

E. The Evaluation

Many of the challenges encountered before and during the evaluation for the speech SF task have already been highlighted earlier in Section IV-A. In particular, the absence of urgency annotations for most speech datasets was an important issue for the speech SF task, and we had to rely on text datasets for urgency labels while training the models.

Lessons Learned

- ASR performance is crucial to this task. If ASR output is poor the errors propagate to the rest of the pipeline.
- However, building a reliable ASR system within 24 hours proved to be a challenging task
- More speech datasets with urgency annotations might be required for a more reliable estimation of urgency from speech. In the absence of such data, we had to rely on models trained on text data which might not be an optimal choice for speech input with noisy ASR and MT systems.
- The incorporation of multilingual system and G2P mapping for this evaluation helped in getting a better initial system

[2]http://service.subasa.info/transliter2.htm

- Using Tamil transilteration and including it with Sinhalese system helped in boosting Sinhalese performance
- The time consumption to prepare a fst graph composing hmm, lexicon and language model modules was more than expected. This is one reason why CP1 submission was not possible.

F. Remaining Challenges

- The pipeline we employ is "fragile". Errors in a component propagate throughout the pipeline and hurt performance. This effect is especially pronounced in time-sensitive checkpoints.
- We are still very dependent on MT performance. We expected to have some MT-independent components for this evaluation, but they never reached the required performance. We are hoping to use them in the upcoming evaluations.

Acknowledgement

This work was supported by the U.S. DARPA LORELEI Program No. HR0011-15-C-0115. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] L. Huang, K. Cho, B. Zhang, H. Ji, and K. Knight, "Multi-lingual common semantic space construction via cluster-consistent word embedding," in *arxiv*, 2018.

[2] Y. Lin, S. Yang, V. Stoyanov, and H. Ji, "A multi-lingual multi-task architecture for low-resource sequence labeling," in *Proc. The 56th Annual Meeting of the Association for Computational Linguistics (ACL2018)*, 2018.

[3] X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji, "Cross-lingual name tagging and linking for 282 languages," in *Proc. the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*, 2017.

[4] B. Zhang, Y. Lin, X. Pan, D. Lu, J. May, K. Knight, and H. Ji, "Elisa-edl: A cross-lingual entity extraction, linking and localization system," in *Proc. NAACL-HLT2018 Demo Track*, 2018.

[5] G. Lample, M. Ballesteros, K. Kawakami, S. Subramanian, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics –Human Language Technologies (NAACL-HLT 2016)*, 2016.

[6] Y. Lin, C. Costello, B. Zhang, D. Lu, H. Ji, J. Mayfield, and P. McNamee, "Platforms for non-speakers annotating names in any language," in *Proc. ACL2018 Demo Track*, 2018.

[7] X. Pan, T. Cassidy, U. Hermjakob, H. Ji, and K. Knight, "Unsupervised entity linking with abstract meaning representation," in *Proc. NAACL-HLT*, 2015.

[8] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL2018*, 2018.

[9] U. Hermjakob, J. May, and K. Knight, "Out-of-the-box universal romanization tool," in *Proc. ACL2018 Demo Track*, 2018.

[10] H. Xu, M. Marcus, C. Yang, and L. Ungar, "Unsupervised morphology learning with statistical paradigms," in *Proc. COLING2018*, 2018.

[11] M. Galley, M. Hopkins, K. Knight, and D. Marcu, "What's in a translation rule?" in *HLT-NAACL 2004: Main Proceedings*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 273–280.

[12] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 961–968. [Online]. Available: http://www.aclweb.org/anthology/P06-1121

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[14] U. Hermjakob, J. May, and K. Knight, "Out-of-the-box universal romanization tool uroman," in *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, 2018, pp. 13–18. [Online]. Available: http://aclweb.org/anthology/P18-4003

[15] L. Rolston and K. Kirchhoff, "Collection of bilingual data for lexicon transfer learning," Technical Report UW-EE-2016-0001, Tech. Rep., 2016.

[16] F. Braune and A. Fraser, "Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 81–89. [Online]. Available: http://dl.acm.org/citation.cfm?id=1944566.1944576

[17] K. Heafield and A. Lavie, "Combining machine translation output with open source: The carnegie mellon multi-engine machine translation scheme," 2010.

[18] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ser. ACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 160–167. [Online]. Available: http://dx.doi.org/10.3115/1075096.1075117

[19] U. Hermjakob, J. May, M. Pust, and K. Knight, "Translating a language you don't know in the chinese room," in *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, 2018, pp. 62–67. [Online]. Available: http://aclweb.org/anthology/P18-4011

[20] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 2005, pp. 363–370.

[21] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *CoRR*, vol. abs/1508.07909, 2015. [Online]. Available: http://arxiv.org/abs/1508.07909

[22] M. Hannemann, J. Trmal, L. Ondel, S. Kesiraju, and L. Burget, "Bayesian joint-sequence models for grapheme-to-phoneme conversion," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2836–2840.

[23] M. Karafiát, L. Burget, F. Grézl, K. Veselý, and J. Černocký, "Multilingual region-dependent transforms," in *International Conference on Acoustics, Speech and Signal Processing ICASSP, 2016*. IEEE, 2016, pp. 5430–5434.

[24] P. Papadopoulos, R. Travadi, C. Vaz, N. Malandrakis, U. Hermjakob, N. Pourdamghani, M. Pust, B. Zhang, X. Pan, D. Lu, Y. Lin, O. Glembek, M. Karthick B, M. Karafiat, L. Burget, M. Hasegawa-Johnson, H. Ji, J. May, K. Knight, and S. Narayanan, "Team ELISA system for DARPA LORELEI speech evaluation 2016," in *In Proceedings of Interspeech*, August 2017.

[25] A. Stolcke, "Srilm –an extensible language modeling toolkit," in *Proceedings of ICSLP*, 2002, pp. 901—904.

[26] A. Jaech and M. Ostendorf, "Leveraging twitter for low-resource conversational speech language modeling," https://arxiv.org/pdf/1504.02490.pdf.

[27] P. F. Brown, V. J. D. Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467—479, 1992.

[28] P. Liang, "Semi-supervised learning for natural language," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.