ELISA System Description for LoReHLT 2017

Leon Cheung, Thamme Gowda, Ulf Hermjakob, Nelson Liu, Jonathan May, Alexandra Mayn, Nima Pourdamghani, Michael Pust, Kevin Knight
Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
knight@isi.edu

Nikolaos Malandrakis, Pavlos Papadopoulos, Anil Ramakrishna, Karan Singla, Victor Martinez, Colin Vaz, Dogan Can, 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

Xiaoman Pan, Boliang Zhang, Ying Lin, Di Lu, Lifu Huang, Kevin Blissett, Tongtao Zhang, 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

Camille Goudeseune, Mark Hasegawa Johnson, Leda Sari, Wenda Chen, Angli Liu*
Beckman Institute 2011
University of Illinois, Urbana, IL, 61801
* P.G. Allen School of Computer Science & Engineering
University of Washington, Seattle, WA 98195
jhasegaw@illinois.edu

I. Tasks, Conditions and Submissions

We participated in all three tasks for text: Entity Discovery and Linking (EDL), Machine Translation (MT) and Situation Frame (SF), and SF for speech, for both incident languages (ILs): IL5 (Tigrinya) and IL6 (Oromo).

Tables I, II, III, IV, V and VI summarize the submissions for each task, condition, and checkpoint.

A. EDL Highlights

Some new and successful approaches for EDL include:

- We developed a multi-lingual common space which allows related languages to share and transfer resources and knowledge on multiple levels.
- We developed various incident-drive knowledge resource acquisition methods to obtain name gazetteers, name translations and contextual words.
- We developed effective text normalization methods and significantly improved IL6 name tagging, linking and clustering.
- We developed a general Chinese Room framework to perform rapid data selection and annotation by non-native speakers.

We will analyze the detailed impact of these techniques in Section II-G, including additional submissions which are not part of the rearranged ensembles.

B. MT Highlights

We used a variety of MT systems, including phrase-based, Hiero, and syntax-based approaches. We built special out-of-vocabulary (OOV) word translators, plus a do-not-translate tagger that kept OOV translation in check. We effectively developed in-domain parallel data with Native Informants, using our Chinese Room interface.

C. SF Highlights

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 developed a system producing status variables for the frames, based on our Sentiment classification system from the LORELEI Sentiment pilot. We used EDL linking information to improve localization performance. 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 and lead to dramatic performance improvements in cases where MT BLEU was low.

II. Entity Discovery and Linking

Our EDL team consisted of Xiaoman Pan, Boliang Zhang, Ying Lin, Di Lu, Lifu Huang, Kevin Blissett, Tongtao Zhang,

TABLE I
ELISA IL5 EDL Rearranged Ensemble Submissions

Ensemble	Check Point	Condition	Submission	Description
1	1	Constrained	1622	Full system with the best components of name tagging, linking and clustering. Name tagger is based on bi-LSTM+CRFs using noisy training data. Linking is based on name translation and collective inference. NIL clustering is based on text normalization.
	2	Constrained	1865	
	3	Constrained	2316	
2	1	Constrained	1622	The same as constrained-1, with an additional within-document coreference resolution component for person name mentions based on heuristic rules in CP3.
	2	Constrained	1865	
	3	Constrained	2303	

TABLE II
ELISA IL6 EDL Rearranged Ensemble Submissions

Ensemble	Check Point	Condition	Submission	Description
1	1	Constrained	1635	Full system with the best components of name tagging, linking and clustering. Name tagger is based on bi-LSTM+CRFs using noisy training data. Linking is based on name translation and collective inference. NIL clustering is based on text normalization.
	2	Constrained	2064	
	3	Constrained	2391	
2	1	Constrained	1635	The same as constrained-1, but increase threshold for linking so more partial match mentions become NIL.
	2	Constrained	2064	
	3	Constrained	2438	
3	1	Constrained	1635	The same as constrained-1, but set different thresholds for linking in CP3.
	2	Constrained	2064	
	3	Constrained	2452	
4	1	Constrained	1635	The same as constrained-1, but set different thresholds for linking in CP3.
	2	Constrained	2064	
	3	Constrained	2374	
5	1	Constrained	1635	The same as constrained-1, but apply a name tagger with higher precision and lower recall model trained from different data split in CP2.
	2	Constrained	2062	
	3	Constrained	2391	
6	1	Constrained	1635	The same as constrained-2, but apply a name tagger with higher precision and lower recall model trained from different data split in CP2.
	2	Constrained	2062	
	3	Constrained	2438	
7	1	Constrained	1635	The same as constrained-3, but apply a name tagger with higher precision and lower recall model trained from different data split in CP2.
	2	Constrained	2062	
	3	Constrained	2452	
8	1	Constrained	1635	The same as constrained-4, but apply a name tagger with higher precision and lower recall model trained from different data split in CP2.
	2	Constrained	2062	
	3	Constrained	2374	
9	1	Constrained	1635	The same as constrained-1, but keep unstranslated singleton NIL entities in CP3.
	2	Constrained	2064	
	3	Constrained	2425	
10	1	Constrained	1635	The same as constrained-5, but keep unstranslated singleton NIL entities in CP3.
	2	Constrained	2062	
	3	Constrained	2425	

Dian Yu, Samia Kazemi, Ulf Hermjakob, Nima Pourdamghani, Kevin Knight and Heng Ji.

A. Core Algorithmic Approach

The overall framework follows our EDL system for 282 languages [1] and consists of three steps: (1) Incident Language (IL) name tagging; (2) Translate IL names to English and link them to English knowledge base (KB); and (3) cluster unlinkable (NIL) name mentions. 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 [2] as our underlying learning model for name tagging. One novel addition to this framework we made this year is character embedding learning. Instead of learning word embeddings directly, for each language, we applied a Convolutional Neural Network over the sequence of characters of each word, and a max-over-time pooling function to compose word representations. For each language, we further optimized each word' s representation with a multi-layer Long Short-term Memory (LSTM) and a softmax function, minimizing the loss between the predicted distribution over next word and the actual next word. This architecture requires a large amount of training data in order to be effective. In the LoreHLT2017 setting we are provided some labeled data for IL5 (140 documents from REFLEX) and no labeled data for IL6. We have developed the following approaches to acquire noisy training data:

RL to IL Converter. We developed a general framework to convert a word in a related language (RL) to a word in an incident language (IL). In LoreHLT2017 we chose RL5 =

Check Point	Condition	Submission	Description
1	Constrained	cp1c1	vanilla sbmt, v2 data, edit-distance oov, pre-found dictionaries
1	Constrained	cp1c2	buggy
1	Constrained	cp1c3	same as c1, but parallel data auto resegmented
1	Constrained	cp1c4	system combination (syscomb) of [cp1] u3,u1,c5, u3 without oov
1	Constrained	cp1c5	u3 rescored with nmt built with transfer learning from french-english (nmt-ch)
1	Constrained	cp1c6	syscomb of u1,u2,u3,u4,u5
1	Constrained	cp1c7	copy of u4
1	Constrained	cp1c8	cp1 setup, but with UW post-edit oov handling, instead of edit distance
1	Constrained	cp1c9	syscomb of u1,u2,u3,u4,u5
1	Unconstrained	cp1u1	moses v3 data
1	Unconstrained	cp1u2	hiero v3 data
1	Unconstrained	cp1u3	sbmt v3 data with amh2tir, edit distance-based oov finder (edoov)
1	Unconstrained	cp1u4	u3 rescored with nmt built without any transfer learning (nmt-sa)
2	Constrained	cp2c1	system combination of cp2c2 cp2c3 cp1u3 and cp2c5
2	Constrained	cp2c2	moses v6 data
2	Constrained	cp2c3	nd nmt v5 data
2	Constrained	cp2c4	cp2u5
2	Constrained	cp2c5	isi nmt v5 data, amh2tir, tgdict5
2	Unconstrained	cp2u1	cp2u2 with postprocessing
2	Unconstrained	cp2u2	cp1u3 with parallel data changed from v3 to v5
2	Unconstrained	cp2u3	cp2u2 +dict v5
2	Unconstrained	cp2u4	cp1u3 + dict v5
2	Unconstrained	cp2u5	cp2u4 + nmt-sa
2	Constrained	cp2u6	cp2u4 + nmt-ch
2	Constrained	cp2u7	source as submission probe
3	Constrained	cp3c1	cp2c4
3	Constrained	cp3c2	cp3u1
3	Constrained	cp3c3	sbmt v6 data, v8 dict, amh2tir, uw oov
3	Constrained	cp3c4	hiero s11
3	Constrained	cp3c5	moses v6
3	Constrained	cp3c6	syscomb of cp3u3, cp3u4, sbmt v6 data amh2tir v8 dict uwoov, cp3c4, cp3c5
3	Constrained	cp3c7	cp3u4 + nmt-sa
3	Constrained	cp3c8	hiero s10
3	Unconstrained	cp3u1	cp2u5 with v3 data instead of v5
3	Unconstrained	cp3u2	syscomb of cp3c5, cp2c3, cp3u1, cp3u1 without nmt-sa
3	Unconstrained	cp3u3	cp3u4 + edoov
3	Unconstrained	cp3u4	sbmt v6 data amh2tir v8dict
3	Unconstrained	cp3u5	oracle sentence merge of cp3u1, cp2u5, cp3u3 (cp3u1 base)
3	Constrained	cp3u6	trained system selector based on per-sentence scores in cp3u1, cp2u5, cp3u3 (buggy)
3	Constrained	cp3u7	cp3c4
3	Constrained	cp3u8	bugfix of cp3u6

TABLE III
ELISA IL5 MT Submissions

Amharic and RL6 = Somali. The converter consists of four steps in the following order: (1) we gather an RL-English lexicon and an IL-English lexicon, and align RL and IL entries using their English translations as anchors. (2) For each of the remaining RL words, we then find its best IL counterpart with the shortest edit distance lower than some threshold. (3) if (2) fails, we try to find its best IL6 counterpart with the soundex similarity higher than some threshold. (4) If all of the above steps fail, we build a common semantic space based on Canonical Correlation Analysis. We map the embeddings of words in RL to the semantic space of IL by computing cross-lingual semantic similarity. Using this approach we converted 78.6% Amharic words to Tigrinya, and 67.1% Somali words to Oromo. Table IX and Table X present some Amharic-to-Tigrinya and Somali-to-Oromo word and name examples converted from each step respectively. As we obtained more noisy training data across check points, the RL to IL converted annotations did not provide significant improvement for EDL, but provided 4 point BLEU gain for IL5 MT in CP2 and about 1 point BLEU gain for IL6 MT in CP3.

Chinese Room. We applied cross-lingual topic modeling based on lexicons to clustered all IL documents in Set 0-2 and English documents in Set S and Leidos corpus, then we selected incident related IL documents based on the keywords listed in the scenario model. We built a "Chinese Room" EDL interface where an IL document is displayed, and words and candidate names are translated based on lexicons and gazetteers. A user can also collect and provide his/her knowledge about an IL in the interface, such as name designators. The romanized version of IL5 is also displayed. This interface allows a user to identify, classify and translate names in each IL sentence. The interface also allows a user to delete a sentence with low annotation confidence. Besides native informants (NIs) provided, our system developers who are non-native speakers also used this interface to generate noisy name annotations.

Entity Linking.

We mined IL-English name translation pairs from various

Check Point	Condition	Submission	Description
1	Constrained	cp1c1	sbmt v2 parallel data
1	Constrained	cp1c2	sbmt v3 parallel data
1	Constrained	cp1c3	sbmt v4 parallel data
1	Constrained	cp1c4	syscomb of sbmt v4 data edoov, sbmt v4 data edoov + nmt-ch
1	Constrained	cp1c5	sbmt + nmt-ch
1	Constrained	cp1c6	syscomb of cp1u4, cp1u3, cp1u1, sbmt v2 dict + nmt-ch
1	Constrained	cp1c7	sbmt v4 data, v2 dict nmt-sa
1	Constrained	cp1c8	cp1u5 + uw oov
1	Unconstrained	cp1u1	hiero s6
1	Unconstrained	cp1u2	cp1u5 + nmt-sa
1	Unconstrained	cp1u3	sbmt v2 dict + edoov
1	Unconstrained	cp1u4	sbmt v2 dict
1	Unconstrained	cp1u5	sbmt 25 hour system
1	Unconstrained	cp1u6	sbmt v4 data, no postproc
2	Constrained	cp2c1	cp1c3 + dict v6
2	Constrained	cp2c2	cp2c1 + spell normalization
2	Constrained	cp2c3	sbmt v4 data + dict v6 + restrictive identity + detect english
2	Constrained	cp2c4	cp1c3
2	Constrained	cp2c5	syscomb of 3 sbmts, hiero, moses
2	Constrained	cp2c6	UW postedit OOV on top of cp1c3
2	Constrained	cp2c7	source as mt probe
2	Constrained	cp2c8	smbt v4 dict v6
2	Unconstrained	cp2u1	hiero s7
2	Unconstrained	cp2u2	hiero s9
2	Unconstrained	cp2u3	sbmt data v4 dict v6
2	Unconstrained	cp2u4	sbmt data v5 dict v4
2	Unconstrained	cp2u5	cp2u4 + ch-nmt
2	Unconstrained	cp2u6	cp2u5 no postproc
2	Unconstrained	cp2u7	cp2u4 no postproc
2	Unconstrained	cp2u8	sbmt data v4 dict v4
2	Unconstrained	cp2u9	moses v6
2	Unconstrained	cp2u10	buggy
3	Constrained	cp3c1	sbmt v6 som2eng dict v7 copyme uw oov
3	Constrained	cp3c2	cp3u1 - oov + maroon
3	Constrained	cp3c3	hiero s11
3	Constrained	cp3c4	moses v6
3	Constrained	cp3c5	sbmt v6 som2eng dict v7 copyme uw
3	Constrained	cp3c6	cp3c1, but attempt to normalize oromo spelling
3	Unconstrained	cp3u1	copyme v1
3	Unconstrained	cp3u2	copyme v2
3	Unconstrained	cp3u3	copyme v3
3	Unconstrained	cp3u4	copyme v5
3	Unconstrained	cp3u5	sbmt v6data som2eng copyme v6 dict v7
3	Unconstrained	cp3u6	copyme v5 + ml
3	Unconstrained	cp3u7	cp3u5 + edit-distance oov
3	Unconstrained	cp3u8	sbmt dict7 copyme v5 + oov word replace
3	Unconstrained	cp3u9	sentence oracle of cp3u5, cp3u8, cp2c7

TABLE IV
ELISA IL6 MT Submissions

approaches: (1) Cross-lingual Wikipedia titles; (2) Cross-lingual Geoname titles; (3) Name translation pairs mined from IL5 parallel sentences by automatically extracting names from English side and manually aligning them with names in IL in the Chinese Room interface; (4) We collected incident-related English names by automatically extracting names from English scenario model documents and Leidos documents, as well as mining all Oromia region names from Geoname database. Then we translated these names based on gazetteers and soundex matching, and asked NIs to translate the remaining ones to IL6. (5) Using these name translation pairs and lexicon as seeds, we adopted a bootstrapping based graph alignment approach [3] to mine more name pairs from comparable documents in Set0-2 and Set S/Leidos. At the end of CP3 we acquired 2,897 IL5-English name pairs and 1,899 IL6-English name pairs.

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. Only 500k out of 4.7 million entities in DBpedia are linked to GeoNames. We associate mentions with entities in the target KB in a collective manner, based on salience, similarity and coherence measures [4]. We calculated topic-sensitive PageRank scores for 500k overlapping entities between GeoNames and Wikipedia as their salience scores. Then we construct a

TABLE V
ELISA IL5 SF Text Submissions. Final ensembles listed.

Check Point	Condition	Submission	ID	Description
1	Constrained	cp1c1	1445	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
1	Constrained	cp1c2	1444	Same as (cp1c1), but also “crime” frames not allowed
1	Constrained	cp1c3	1445	Same as (cp1c1)
1	Constrained	cp1c4	1446	MLP-LSA
1	Constrained	cp1c5	1447	CNN-GRU
1	Constrained	cp1c6	1458	Combination: MLP-LSA + LEIDOS
1	Constrained	cp1c7	1451	Combination: MLP-LSA + CNN-GRU + LEIDOS
1	Constrained	cp1c8	1486	baseline model
1	Constrained	cp1c9	1456	HATT model
1	Constrained	cp1c10	1456	same as (cp1c9)
2	Constrained	cp2c1	1834	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
2	Constrained	cp2c2	1833	Same as (cp2c1), but also “crime” frames not allowed
2	Constrained	cp2c3	1830	Same as (cp2c1), but “regime” frames allowed
2	Constrained	cp2c4	1835	MLP-LSA
2	Constrained	cp2c5	1836	CNN-GRU
2	Constrained	cp2c6	1839	Combination: MLP-LSA + LEIDOS
2	Constrained	cp2c7	1838	Combination: MLP-LSA + CNN-GRU + LEIDOS
2	Constrained	cp2c8	1849	baseline model
2	Constrained	cp2c9	1824	HATT model
2	Constrained	cp2c10	1846	HATT v2 model, with one extra layer over (cp2c9)
3	Constrained	cp3c1	2270	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
3	Constrained	cp3c2	2271	Same as (cp3c1), but also “crime” frames not allowed
3	Constrained	cp3c3	2265	Same as (cp3c1), but “regime” frames allowed
3	Constrained	cp3c4	2266	MLP-LSA
3	Constrained	cp3c5	2267	CNN-GRU
3	Constrained	cp3c6	2268	Combination: MLP-LSA + LEIDOS
3	Constrained	cp3c7	2269	Combination: MLP-LSA + CNN-GRU + LEIDOS
3	Constrained	cp3c8	2264	baseline model
3	Constrained	cp3c9	2245	HATT model
3	Constrained	cp3c10	2256	HATT v2 model, with one extra layer over (cp3c9)

TABLE VI
ELISA IL6 SF Text Submissions. Final ensembles listed.

Check Point	Condition	Submission	ID	Description
1	Constrained	cp1c1	1441	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
1	Constrained	cp1c2	1439	Same as (cp1c1), but also “crime” frames not allowed
1	Constrained	cp1c3	1441	Same as (cp1c1)
1	Constrained	cp1c4	1448	MLP-LSA
1	Constrained	cp1c5	1449	CNN-GRU
1	Constrained	cp1c6	1459	Combination: MLP-LSA + LEIDOS
1	Constrained	cp1c7	1450	Combination: MLP-LSA + CNN-GRU + LEIDOS
1	Constrained	cp1c8	1487	baseline model
1	Constrained	cp1c9	1463	HATT model
1	Constrained	cp1c10	1463	same as (cp1c9)
2	Constrained	cp2c1	1843	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
2	Constrained	cp2c2	1844	Same as (cp2c1), but also “crime” frames not allowed
2	Constrained	cp2c3	1831	Same as (cp2c1), but “regime” frames allowed
2	Constrained	cp2c4	1840	MLP-LSA
2	Constrained	cp2c5	1841	CNN-GRU
2	Constrained	cp2c6	1842	Combination: MLP-LSA + LEIDOS
2	Constrained	cp2c7	1845	Combination: MLP-LSA + CNN-GRU + LEIDOS
2	Constrained	cp2c8	1850	baseline model
2	Constrained	cp2c9	1825	HATT model
2	Constrained	cp2c10	1847	HATT v2 model , with one extra layer over (cp2c9)
3	Constrained	cp3c1	2276	Combination: MLP-LSA + CNN-GRU, no “regime” frames allowed
3	Constrained	cp3c2	2277	Same as (cp3c1), but also “crime” frames not allowed
3	Constrained	cp3c3	2260	Same as (cp3c1), but “regime” frames allowed
3	Constrained	cp3c4	2272	MLP-LSA
3	Constrained	cp3c5	2273	CNN-GRU
3	Constrained	cp3c6	2275	Combination: MLP-LSA + LEIDOS
3	Constrained	cp3c7	2274	Combination: MLP-LSA + CNN-GRU + LEIDOS
3	Constrained	cp3c8	2257	baseline model
3	Constrained	cp3c9	2244	HATT model
3	Constrained	cp3c10	2243	HATT v2 model, with one extra layer over (cp3c9)

TABLE VII
ELISA IL5 SF Speech Submissions. Final ensembles listed.

Check Point	Condition	Submission	ID	Description
1	Constrained	cp1c1	2513	MLP with relevance classifier(RC) and BUT ASR
1	Constrained	cp1c2	2514	CNN-GRU with relevance classifier and BUT ASR
1	Constrained	cp1c3	2515	MLP + CNN-GRU with relevance classifier and BUT ASR
1	Constrained	cp1c4	2518	CNN-GRU without relevance classifier and BUT ASR
1	Constrained	cp1c5	2519	MLP + CNN-GRU without relevance classifier and BUT ASR
1	Constrained	cp1c6	2520	CNN-GRU with relevance classifier and UIUC ASR
1	Constrained	cp1c7	2522	MLP + CNN-GRU without relevance classifier and BUT ASR. Different training set
1	Constrained	cp1c8	2523	CNN-GRU without relevance classifier and BUT ASR. Different training set
1	Constrained	cp1c9	2524	Combination of two MLP + CNN-GRU systems, one with RC the other without. BUT ASR
1	Constrained	cp1c10	2526	MLP without relevance classifier and BUT ASR

TABLE VIII
ELISA IL6 SF Speech Submissions. Final ensembles listed.

Check Point	Condition	Submission	ID	Description
1	Constrained	cp1c1	2496	MLP without relevance classifier(RC) and BUT ASR
1	Constrained	cp1c2	2498	CNN-GRU without relevance classifier and BUT ASR
1	Constrained	cp1c3	2499	MLP + CNN-GRU with relevance classifier and BUT ASR
1	Constrained	cp1c4	2500	MLP without relevance classifier and UIUC ASR
1	Constrained	cp1c5	2501	Combination of two MLP systems without RC, one using BUT ASR the other UIUC ASR
1	Constrained	cp1c6	2502	MLP with relevance classifier(RC) and BUT ASR
1	Constrained	cp1c7	2503	MLP + CNN-GRU with relevance classifier and BUT ASR.
1	Constrained	cp1c8	2504	MLP without relevance classifier and BUT ASR. Different training set
1	Constrained	cp1c9	2505	MLP with relevance classifier and BUT ASR. Different training set
1	Constrained	cp1c10	2506	MLP + CNN-GRU without relevance classifier and BUT ASR

TABLE IX
RL (Amharic) to IL (Tigrinya) Conversion Examples

Conversion Method	Amharic	Tigrinya
Lexicon	**ዩናይትድ ስቴትስ (United States)**	**ሕቡራት መንግስታት አሜሪካ**
Edit Distance	**በደቡብ ሱዳን**	**ንደቡብ ሱዳን**
	በካሊፎርኒያ	**ካሊፎርኒያ**
	በፓስፊክ	**ፓስፊክ**
	የሶማሊያ	**ሶማሊያ**
	የአፍሪካ	**አፍሪካ**
Embedding	**መቀሌ (Mekele)**	**መቐለ**

TABLE X
RL (Somali) to IL (Oromo) Conversion Examples

Conversion Method	Somali	Oromo
Lexicon	Hayyuudha Garoomaadha Madobee (Oromo Democratic Front)	Kallacha Walabummaa Oromiyaa
Edit Distance	Somaaliilandi	Somaaliiland
	Indooneesiyaa	Indooneeshiyaa
	Filiipins	Filippiinsi
	Angeelaa Merkeel	Angelaa Merkel
	Gaalkaayyoo	Galkaayoo
Soundex	Guinea	Giinii
	Hargiisa	Hargeessaa
	Tanzania	Tanzaniyaa
	Melbourne	Meelboorn
	Queensland	Quwiinsilaandi
Embedding	CID (Coalition for Unity and Democracy)	CUD

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.

NIL Clustering.

NIL clustering is especially challenging for IL6 due to the numerous spelling variants for each word. For example, there are about 244 different spellings in Set 0-2 for the entity "*Ethiopia*". 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. We grouped mentions with the same normalized surface form, e.g., "*Finfinnee*" and "*magaalaa Finfinne*"; (2) We clustered mentions with similar NYSIIS representation (similar to Soundex) longer than four letters, after removing double consonants and vowels. For example, "*Aanaa Mi□essoo*" and "*Aanaa Muneessaa*" were grouped into one cluster. (3) We clustered two mentions if the edit distance between their normalized surface forms is equal to or smaller than a threshold D, where D = length(mention1) / 8 + 1, which means we allow one character to be different per 8 characters. For example, "*Aanaa Muneessaa*" and "*Anaa Muneesaa*" are grouped. Finally we merged two clusters if they include mentions sharing the same English translation.

B. Critical Additional Features and Tools Used

The name tagging features we exploited for both ILs include: (1) Character embeddings and word embeddings learned from all Set 0-2 monolingual corpora; (2) Name designators translated from English; (3) hierarchical Brown clusters; (4) Stemming features based on various morphological analysis methods as described in Section II-D below; and (5) GPE/LOC names in ILs collected from Geoname which is part of the KB. For IL5 we trained a part-of-speech (POS) tagger from REFLEX data and used POS tags as additional features. We asked NIs to translate English names of Oromia regions to IL6 and used them as additional features for IL6. We also used the fabulous universal romanizer developed by ISI in our annotation interface to create noisy training data from Set0-2.

C. Other Data Used

We used multi-lingual Wikipedia and DBPedia dumps, and massively multi-lingual Panlex which were collected before the evaluation.

D. Significant Pre/Post-Processing

Morphology analysis is very important for both ILs, especially to normalize many spelling variants of Oromo words. We exploited both UPenn and JHU morphology analyzers but they didn't provide positive gains in the evaluation sets, though UPenn morphology analysis provided 0.7% improvement on IL5 development set from Set1.

Therefore we developed our own IL6 text normalizer to learn more generalizable word embeddings, and used normalized forms to perform better name translation matching in linking and NIL clustering. We generated all possible inflections for nouns and adjectives, including Noun plural, Definiteness, Nominative case, Genitive case, Dative case, Instrumental case, Locative case, Ablative case, Feminie adjective, Adjective plural, and replaced/fixed consonant combinations (e.g., bt ==> bd), And (-f), and Also (-s). We then used these variants to map each token to its most popular counterpart, its shortest counterpart, and its stemming form by further removing, duplicating or shortening the ending vowels. For IL6 the vocabulary size is reduced from 234,001 to 100,166 after normalization.

We developed a special post-processing step for @ mentions and hashtags in tweets, by automatically parsing each mention into multiple tokens, and running English EDL to candidate names.

E. Native Informant Use

We asked the Native Informant to perform the following tasks at each check point:

- CP1: Translated incident related English names automatically extracted from Leidos Corpora into IL5 (failed due to lack of efficient typing software) and IL6.
- CP2: Annotated some incident related documents; Translated English names of Oromia regions into IL6; Translated system extracted most frequent IL names in Set1 into English; answered some linguistic questions such as the meaning of affixes.
- CP3: Annotated some incident related documents; Translated system extracted most frequent IL names in Set2 into English; Translated incident related English names automatically extracted from Set S into IL6.

F. EDL for Speech

We apply the best CP3 text EDL system to Automatic Speech Recognition (ASR) output for both ILs. For IL6 name tagging we applied the model trained from lower-cased data because ASR output is not truecased. We applied the best CP3 text EDL system to Set0-2, along with English names extracted from Set S, and constructed a "expect-to-appear" name gazetteer that includes IL name, its English translation if it's available, and its frequency in Set0-2. BUT and UIUC-Speech teams used this name gazetteer for their LM adaptation so ASR can recognize these names (see Sec. V). We also applied BUT's English and IL keyword spotting (KWS) algorithms to directly search these expected IL names and their English translations in IL speech. In this way we hope to overcome some ASR missing errors on names. Unfortunately we obtained too many matched candidates in speech and did not get enough time to work around various thresholds.

G. Evaluation Results and Analysis

In this section we will present the evaluation results and the impact of some new and successful methods for each checking point. We report the overall end-to-end CEAF results (EDL), name tagging (NER), and name tagging and linking (NEL).

Checkpoint 1.

For IL5, we used 5,003 name annotated sentences from LDC REFLEX corpus and 613 sentences from Chinese Room annotated by non-native speakers to train the name tagger.

IL5	Description	EDL	NER	NEL
1622	Best Run	**0.425**	**0.723**	**0.450**

For IL6 we 2,382 sentences from Chinese Room annotated by non-native speakers to train the name tagger. We noticed that there are many capitalized tokens in Set 0 tweets, so we decided to lower-case all training data and trained a separate model for tweets. This provides 6.2% name tagging F-score gain. We also applied the tagger trained from lower-cased data to process Automatic Speech Recognition (ASR) output for the speech localization evaluation.

IL6	Description	EDL	NER	NEL
1634	no Lowcase for SN	0.156	0.295	0.158
1635	Best Run	**0.189**	**0.357**	**0.195**

We learned two major lessons from CP1: (1) Our IL6 name tagger achieved 81% F-score on a development set selected from Set 0. This huge gap between the development set and the evaluation set indicated that we should improve our development set selection based on the scenario model. (2) We noticed the spelling variant problem in IL6 but did not get time to develop a text normalizer. So we decided to focus on it in CP2.

Checkpoint 2.

After applying hashtag parser and English EDL, we obtained 0.3% improvement in IL5 name tagging (run 1667). For IL5 Up to CP2 we obtained 813 sentences from Chinese Room annotated by both non-native speakers and NIs. The new training data provided 2.1% improvement on name tagging. New name translation pairs significantly improved linking and clustering.

IL5	Description	EDL	NER	NEL
1667	Best CP1 + hashtag	0.403	0.726	0.426
1865	Best Run (1667 + more name translation)	**0.575**	**0.747**	**0.628**

We read the IL6 entity annotation guideline more carefully and noticed that if a group of people is involved in a social movement it should be tagged as PER, otherwise it's not a name. From Set 0 and Set 1 we noticed that "*oromo*" appears very often, so we did special hashtag processing by tagging hashtags like "*#oromoprotest*" and "*#oromorevolution*" as PER but remove all "*#oromo*" from name candidates. This provided 6.8% absolute name tagging F-score improvement (run 1687). Then we propagated all Oromia region names translated from English to IL6 by the NIs and obtained 6.6% further improvement on name tagging (run 1709). Up to cp2 we obtained 6,093 sentences from Chinese Room annotated by both non-native speakers and NIs. The new name tagger trained from the new data set provided 10.1% improvement (run 1731). We used lexicon translation as feedback to remove low-frequency names with all lower-case translations and obtained 1.8% improvement (run 1733). Following the guideline we tagged more slogan abbreviations that include people who participated in protests as PER (e.g., FDG, FXG) and obtained 3% more improvement (run 1735). The remaining linking and clustering improvement was obtained from more name translation pairs and text normalization. The text normalization component was a major contributor to the big jump of linking and clustering performance from CP1 to CP2, because it effectively clustered name variants into entities and linked them to the right entries in the KB. Figure 1 shows the impact of text normalizer at "cutting down" the long tail of entity clusters (i.e., reduced the number of singleton entity clusters).

IL6	Description	EDL	NER	NEL
1687	Best CP1 + Hashtag	0.191	0.425	0.237
1709	1687 + Oromia	0.201	0.491	0.247
1731	1709 + more data	0.214	0.592	0.274
1733	1731 + translation feedback	0.227	0.610	0.288
1735	1733 + slogan	0.230	0.640	0.329
2064	Best Run 1 (1735 + more name translation + text normalization)	0.409	**0.679**	0.513
2062	Best Run 2 (1735 + more name translation + aggressive text normalization)	**0.420**	0.664	**0.520**

Checkpoint 3.

We automatically detected English texts in IL5 and directly applied English EDL on them, and obtained 0.8% overall improvement (run 2099). At the end we obtained 4,459 sentences annotated from both non-native speakers and NIs, but compared to CP2, the new annotations did not provide any further significant improvement. One possible reason is that during CP3 non-native speakers did annotations in a rush, and perhaps we have captured easy cases but still failed to find difficult cases without being able to understand the whole context. Text normalization was not so effective for IL5, so some common name mentions were mistakenly identified as NILs, so we decided to apply cluster all mentions first, and then link each cluster to the KB by propagating KB IDs across members in each cluster. This strategy provided 2.6% improvement in name tagging and linking (run 2161). We cleaned the mapping between Wikipedia titles and LORELEI KB titles and obtained 1.1% further improvement in name tagging and linking (run 2170). Finally we added more name translation pairs from NIs and obtained 1.9% overall improvement for the best run.

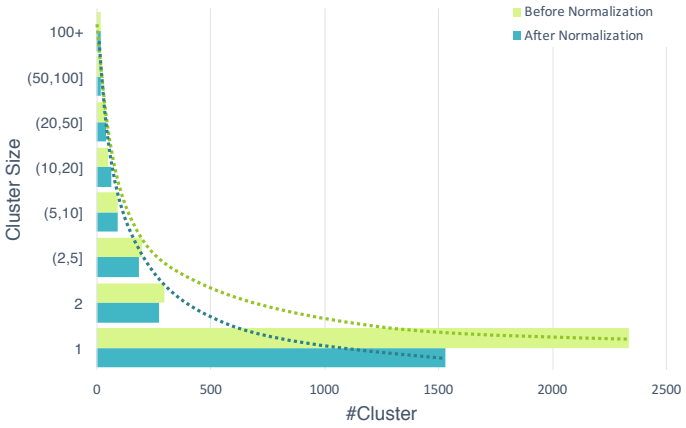

Fig. 1. Impact of Text Normalization on Reducing Singleton Entity Clusters (cutting-down entity long tail)

IL5	Description	EDL	NER	NEL
2099	Best CP2 + English EDL	0.583	0.747	0.635
2161	2099 + Clustering before Linking	0.608	0.747	0.661
2170	2161 + clean KB mapping	0.619	0.747	0.672
2316	Best Run (2170 + more name translation)	**0.638**	**0.748**	**0.690**

We removed some social movement names like "*qeerroon*" from ORG candidates and obtained 1.3% name tagging improvement (run 2111). We cleaned name translations and obtained improvement in name tagging and linking (run 2136). Up to CP3 we obtained annotations for 7,693 sentences from both non-native speakers and NIs. Adding new training data provided 1.4% improvement in name tagging (run 2164). We also attempted to remove all singleton NILs for which we were not able to translate, and obtained 1.2% overall improvement (run 2452). Ignoring east/west/north/south modifiers in GPE/LOC linking provided 0.5% gain in name tagging and linking in the final best run.

IL6	Description	EDL	NER	NEL
2111	Best CP2 (2064) + remove movement	0.442	0.692	0.551
2136	2111 + clean name translation	0.460	0.695	0.558
2164	2136 + more data	0.476	0.709	0.578
2452	2164 + remove untranslated singleton NILs	0.488	**0.713**	0.598
2391	Best Run (2452 + ignoring GPE/LOC modifiers in linking)	**0.490**	0.712	**0.603**

Hill Climbing Summary

Figure 2 and Figure 3 summarize the above EDL hill-climbing stories.

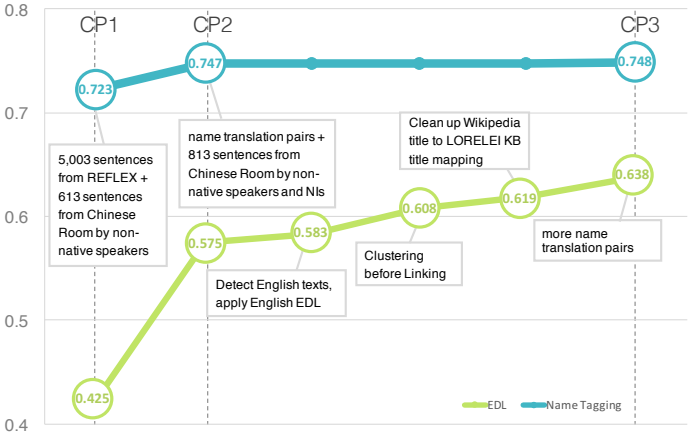

Fig. 2. IL5 EDL Hill Climbing

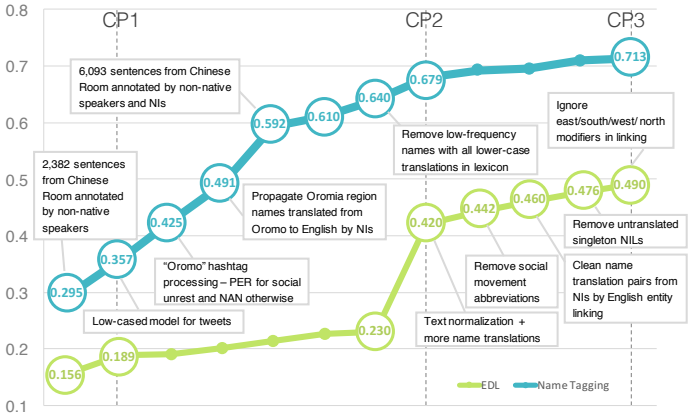

Fig. 3. IL6 EDL Hill Climbing

H. Remaining Challenges

For IL5 we a good starting point for name tagging at CP1 thanks to the REFLEX annotations. However, it was difficult to further boost the performance by adding more annotations by non-native speakers through the Chinese Room, possibly because all low-hanging fruits were already picked, while the limited coverage of lexicon and automatic romanized form did not provide non-native speakers enough support to achieve high recall (CP3 recall is 70.1% and precision is 80.1%). We developed an automatic IL-English parallel sentence mining method but failed to discover near-to-parallel sentence pairs from Set0-2 and Set S/Leidos. Various types of linguistic features such as POS tagging (trained from REFLEX data) and UPenn morphology analysis achieved significant gains in our development set from Set0-2 but hurt the performance on the evaluation set.

For IL6 we were able to bring up the embarrassingly low name tagging performance (35.7%) in CP1 to (71.2%, almost doubled), by keeping adding incident related resources and solving the spelling variant problem. To further investigate the domain mismatch problem, we also ran our "Time 0" system, the 282 languages universal EDL system [1] on two ILs without any tuning, and only obtained 2% name tagging f-scores. We have achieved great success at text normalization which significantly improved name tagging and linking (from 19.5% in CP1 to 60.3% in CP3) and NIL clustering (overall EDL from 18.9% in CP1 to 49.0% in CP3). However the spelling variant problem is not completely solved. Compared to IL5, IL6 has a much longer long tail in the EDL output (about 1,000 more NIL singletons).

Our final CP3 system achieved great linking accuracy for both languages - 92.2% for IL5 and 84.7% for IL6. Addressing most of the remaining errors requires deep background knowledge discovery from English Wikipedia and large English corpora. Some examples as follows.

- *Before 2000, the regional capital of Oromia was Addis Ababa, also known as "Finfinne"*. It's from the text description of "*Oromia Region*" entry in Wikipedia, which teaches us "*Finfinne*" can be linked to "*Addis Ababa*" in the KB.
- *The armed Oromo units in the Chercher Mountains were adopted as the military wing of the organization, the Oromo Liberation Army or OLA*. It's from the text description of "*Oromo Liberation Front*" entry in Wikipedia, which teaches us "*WBO (Oromo Liberation Army)*" is part of "*ABO (Oromo Liberation Front)*" and thus they refer to two different entities.
- The names of the same region may have got frequently changed in the history. The same name mention may refer to different entities at different time points. For example, the Wikipedia entry for "*Jimma Horo*" teaches us that *Jimma Horo may refer to the following: Jimma Horo, East Welega, former woreda (district) in East Welega Zone, Oromia Region, Ethiopia; Jimma Horo, Kelem Welega, current woreda (district) in Kelem Welega Zone, Oromia Region, Ethiopia*. So we would really need to figure out what kind of events and situations these mentions were involved, at what time, in order to be able to correctly linking and clustering them. This is even the major challenge that the state-of-the-art English entity linking is still facing.
- *EPRDF = OPDO + ANDM + SEPDM + TPLF* because of the following facts described in Wikipedia articles:
  – *EPRDF*: Ethiopian People's Revolutionary Democratic

Front, also called *Ehadig*.
  - *OPDO*: Oromo Peoples' Democratic Organization.
  - *ANDM*: Amhara National Democratic Movement.
  - *SEPDM*: Southern Ethiopian People's Democratic Movement
  - *TPLF*: Tigrayan People's Liberation Front, also called *Weyane* or *Second Weyane*, perhaps because there was a rebellion group called Woyane/Weyane in the Tigray province in 1943.
- *Qeerroo* is not an organization although it has its own website, based on what's described in news articles:
  - *The overwhelming belief is that its leaders are hand-picked by the TPLF puppet-masters, and the new generation of Oromo youth –known as the 'Qeerroo' –have seen that it is business as usual after the latest reform.*
  - *The Qeerroo, also called the Qubee generation, first emerged in 1991 with the participation of the Oromo Liberation Front (OLF) in the transitional government of Ethiopia. In 1992 the Tigrayan-led minority regime pushed the OLF out of government and the activist networks of Qeerroo gradually blossomed as a form of Oromummaa or Oromo nationalism.*
  - *Today the Qeerroo are made up of Oromo youth. These are predominantly students from elementary school to university, organising collective action through social media. It is not clear what kind of relationship exists between the group and the OLF. But the Qeerroo clearly articulate that the OLF should replace the Tigrayan-led regime and recognise the Front as the origin of Oromo nationalism.*
- Text normalization mistakenly grouped "*Somali (Somali region)*", "*Somalia*" and "*Somaliland*" but they refer to three different entities:
  - *The Ethiopian Somali Regional State (Somali: Dawlada Deegaanka Soomaalida Itoobiya) is the easternmost of the nine ethnic divisions (kililoch) of Ethiopia.*
  - *Somalia, officially the Federal Republic of Somalia(Somali: Jamhuuriyadda Federaalka Soomaaliya), is a country located in the Horn of Africa.*
  - *Somaliland (Somali: Somaliland), officially the Republic of Somaliland (Somali: Jamhuuriyadda Somaliland), is a self-declared state internationally recognised as an autonomous region of Somalia.*

III. Machine Translation

Our MT team consisted of Leon Cheung, Thamme Gowda, Kenton Murray, Toan Nguyen, Ulf Hermjakob, Nelson Liu, Jonathan May, Alexandra Mayn, Nima Pourdamghani, Michael Pust, Heng Ji, David Chiang, and Kevin Knight.

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 neural system built at Notre Dame.
- A neural system (Neural) based on the Zoph_RNN toolkit from ISI.

These MT systems were trained on parallel data provided by NIST. Translation models and language models used mixed-case data. Word alignment used stemmed corpora (both source and target), down to the first four letters of each word. We used two word aligners (GIZA and Berkeley).

Morphology. Most MT systems used the *Penn* and *Morfessor* systems for unsupervised splitting of words. The SBMT translator was additionally exposed to full-word analyses in a source-language lattice.

Parallel Corpora. We processed the provided parallel data into sets called 'train' (for MT rule acquisition), 'dev' (for MT tuning), and 'test' (held out). We also numbered our parallel data releases (v1, v2, v3, …). In Tigrinya v4 and Oromo v3, we used the Gargantua sentence alignment tool [5], with a modification to use verse numbers as a hint. In Tigrinya v5 and Oromo v4–5, we noted that many Oromo documents were Bible chapters that had been paired with the wrong English Bible chapters; we retrieved the correct chapters and manually sentence-aligned them. Finally, in v6, we noted that even correctly-paired English Bible chapters were from the archaic King James Version; we replaced these with the modern New World Translation from the same website.

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 numbered our dictionary releases (v1, v2, v3, …). For IL5, dictionary sizes were 19,666 (v1), 19,691 (v2, 44,264 (v3), and 85,064 (v4). For IL6, dictionary sizes were 31,178 (v1), 30,039 (v2), 49,435 (v3 and v4).

System combination. We used Kenneth Heafield's Multi-Engine Machine Translation (MEMT) software [6] 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 [7]. We tuned system combination using the 'dev' set and chose systems with high individual Bleu scores.

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 Tigrinya in Latin script.

For the 2016 evaluation, we developed a post-processing cleanup system for Uighur/English SBMT. In the 2017 evaluation, this system hurt MT accuracy (11.21 to 10.41 on Oromo/English), so we removed it.

Out-of-vocabulary word translation. We developed an IL OOV translator that finds a similar known IL word and borrows its English translation, where similarity is done with edit distance. Moreover, we developed a method of extracting a standalone OOV dev/test set by pulling one-count items from the parallel text. For Tigrinya, we obtained 12.9% exact translation accuracy on the test set.. Moving from syllabic script to uromanized script lifted OOV exact-match translation accuracy from 12.9% to 13.7%. Oromo OOV translation accuracy was 22.7%.

We also used an OOV system from UW at postedit time. It helped in Tigrinya/English (12.79 to 13.26), but it hurt in Oromo/English (9.80 to 6.16 Bleu).

Related Languages. We also converted our Amharic/English parallel data to Tigrinya/English via a script developed at RPI (Section II-A), and used it to supplement our existing Tigrinya/English parallel data for MT training. This improved Tigrinya/English MT (12.79 to 16.76).

Re-scoring. Our neural MT (NMT) systems did not perform well in standalone mode, but we profitably used them to re-score SBMT n-best lists. Bleu on Tigrinya/English went from 12.79 to 13.56, and on Oromo/English from 10.41 to 11.06.

System combination. For IL5, system combination improved Bleu from 13.56 to 14.77. However, once a single system reached 16.82, we could not improve it via adding other systems; system combination dropped Bleu from 16.82 to 16.64.

Native Informant (CP1, Tigrinya, one hour): We asked whether 12 sample sentence pairs from our parallel data were actually translations or not. This went fairly slowly. We also asked the NI to orally translate several sentences with us typing the English, and the NI providing oral edits.

Native Informant (CP1, Oromo, one hour): We asked whether sample sentence pairs from our parallel data were actually translations or not. We also asked the NI to orally translate several sentences with us typing the English, and the NI providing oral edits.

Chinese Room: We previously developed a interface (the "Chinese Room") 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. Limited Oromo dictionary resources and interleaved Tigrinya morphology made it challenging to instantly bring up Chinese Room instances for these languages.

Checkpoint 1 submission results.

IL5/Tigrinya	Description	CP1 Bleu
CP1c1	24 hours, v3 data, edOOV	12.79
CP1c5	CP1c1 + NMT re-score	13.56
CP1u3	CP1c1 + Amh/Eng→Tig/Eng	16.76
CP1c7	CP1c5 + CPu3	**16.82**

IL6/Oromo	Description	CP1 Bleu
CP1c1	24 hours, v2 data	9.80
CP1c3	CP1c1 + v4 data	**11.21**
CP1u5	CP1c3 + post-processing	10.41
CP1c5	CP1u5 + post-proc + NMT re-score	11.06

Lessons Learned:

- It was difficult to obtain resources good enough to support the Chinese Room.
- Morfessor morphology worked better in CP1 than Penn morphology.
- We did not get to use dictionary versions v2 or v3 in Tigrinya/English MT.
- Dictionary v2 hurt in Oromo/English MT (10.41 -> 8.38). We found it difficult to analyze why.

Checkpoint 2.

Parallel Data and Dictionaries. We continued to clean our parallel data and expand our dictionaries, including IL/English name-pair lists from IE researchers at RPI.

Related languages. We continued using our Amharic/Tigrinya converter, and we began developing a Somali/Oromo converter. The latter was much more difficult due to the distance between the languages.

In-domain data. We developed small domain parallel sets using the Native Informants, partially by working inside the Chinese Room with them.

System combination. We got a small boost on Oromo/English MT from system combination (11.21 to 11.30).

Doing nothing. After failing to get CP2 improvements on Oromo/English, we finally submitted an output that just copied the Oromo source, and got a Bleu of 11.29, which was surprisingly higher than our best CP1 system, and almost the same as our best CP2 system. We were clearly translating many words that should be left alone. Interestingly, the do-nothing system was worse than our MT system on Meteor (0.147 versus 0.171) and also underperformed our MT system when we counted how many segments were translated better according to Bleu (168 versus 278), as reported by the NIST feedback interface. Nevertheless, by the end of CP2 we realized than any gains from correct MT were offset by over-translating "Oromo" words that should not have been touched.

Native Informant (CP2, Tigrinya, first hour): We did 35 minutes of detailed Chinese Rooming with the NI. Going word by word enabled us to learn about Tigrinya morphology. We spend five minutes on a question of lexicon cleaning/segmentation. The last 20 minutes consisted of free translation of an article about spies and a late-night arrest. The NI was able to translate 22 sentences with our help.

Native Informant (CP2, Tigrinya, second hour): We spent 15 minutes to translate 4 tweets, 15 minutes to translate 10 sentences from the above new article. In the remaining 30 minutes, we translated 10 sentences word-by-word in the Chinese Room interface.

Native Informant (CP2, Tigrinya, third hour): We translated 15 sentences (bombing and refugee articles) using the method

of "NI talks, we type."

Native Informant (CP2, Tigrinya, fourth hour): We spent 45 minutes to finish up a Chinese Room document, plus 10 minutes free translation of a news story.

Native Informant (CP2, Oromo, first and second hour): We were able to completely go through 15 sentences word-by-word in the Chinese Room, obtaining word glosses and understanding morphology, in addition to full translation. We also separately translated Oromo sentences in a plaintext interface.

Native Informant (CP2, Oromo, third hour): We spent 20 minutes doing direct translation of 11 in-domain tweets. We spent 40 minutes going word-by-word in the Chinese room, translating 9 sentences.

Native Informant (CP2, Oromo, fourth hour): We spent time doing direct translation of Oromo texts.

Chinese Room: We added functionality to the Chinese Room, but did not release it for use during CP2, due to the difficulties of low resources—even well-acquainted users could not yet accurately translate inside the Chinese Room.

Checkpoint 2 submission results.

IL5/Tigrinya	Description	CP2 Bleu
CP1c7	Best CP1	16.82
CP2u1	no post-proc	15.70
CP2u2	yes post-proc	16.05
CP1u3	v3 data	16.76
CP2u2	v5 data	16.05
CP2u2	v3 dictionary	16.05
CP2u3	v5 dictionary	16.71
CP2u4	v5 dict	16.87
CP2c4	v5 dict + v5 data + NMT	**17.54**
CP2c1	system combination	17.21

IL6/Oromo	Description	CP2 Bleu
CP1c3	Best CP1	11.21
CP2u7	v5 parallel data	10.71
CP2c5	v? dictionary	10.70
CP2c7	v4 dictionary	11.07
CP2c6	CP1c3 + NMT re-score	11.17
CP2c5	system combination	**11.30**
CP2c7	Oromo source untouched	11.29

Lessons Learned.

- Cleaning up parallel texts and dictionaries for Tigrinya/English continued to improve Bleu. Our v5 dictionary led to better Bleu than our v3 dictionary; however, v5 parallel data led to worse Bleu than v3 parallel data. Data versioning helped us navigate the experimental terrain.
- Submitting source Oromo text (untouched) yielded a better Bleu score on evaluation data than anything we previously tried using MT. This meant that Set E had to contain may tokens that should be passed through rather than translated.

Checkpoint 3.

Related languages. RPI developed a Somali/Oromo converter (Section II-A), and we used this to turn Somali/English data into augmentation for our Oromo/English parallel data.

Do-not-translate (DNT) tagger. We built a CRF tagger to identify IL tokens that should not be translated. This tagger was trained on Somali/English (and other parallel) data, where the foreign side has a "natural" free source of tags—if the foreign word can be found on the English side, we mark it as *do-not-translate* (DNT). This is similar to the *transliterate-me* tagger in [8]. The tagger is given features based on word spelling and context.

To evaluate the DNT tagger directly, we modified Oromo test data by protecting tagged words, and replacing all others by the token "the". We hoped that this would not be much worse than the 11.29 Bleu obtained by submitting Oromo source text as is. (I.e., any occasional failures to protect DNT tokens would be offset by gains from matches on the word "the"). However, the result was 9.33, meaning that we failed to protect many tokens. We overtranslated those, resulting in a significant loss of 2 Bleu points over just submitting Oromo source untouched. We then built another DNT tagger with heuristics derived from manual Set 1 word-type analysis (other words replaced by "the"), and the Bleu score rose to 10.08.

We felt it was important to capture more DNT words, in part because of the low accuracy of MT and OOV translation. If we failed to protect a DNT word, we would need to accurately translate several non-DNT words to balance that loss.

In fact, MT could even do even more harm than translate a word inaccurately. When we analyzed our MT system's workings, we noticed additional two sources of error: (1) Morfessor would frequently break up tokens that should not be translated, allowing the the word pieces to be translated, and (2) even if all foreign word tokens were correctly "passed through" (untouched), MT could freely reorder them or insert function words between them. In such cases, Bleu 4-gram matches could be seriously disrupted. This further explained why we had a hard time beating the "submit Oromo untouched" baseline. We therefore protected DNT tokens from Morfessor, and we joined sequences of consecutive DNT tokens into single massive DNT phrases.

We also created a heuristic, manually-designed DNT tagger (v5). Tokens were protected if they had any character other than ASCII letters and apostrophe, if they look like laughter (e.g., "hahaha"), or contain several key patterns that match Twitter elements, if they are in an English dictionary but not a high-frequency Oromo dictionary, or if they are one of a set of frequently occurring incident- and news-related acronyms. Additionally, tokens that begin with these acronyms were translated as the acronyms themselves (inflection removed). Finally, if 75% or more tokens in a segment were protected, all tokens in the segmentwere protected. This resulted in 20-28% of tokens being protected, based on Sets 1 and 2.

DNT tagger v5, combined with v6 data, v7 dictionary, and Somali/English data converted to Oromo/English by RPI, together yielded an improved Bleu score of 13.21, two points higher than submitting Oromo untouched.

Special treatment of URLs and hashtags/mentions. For the Hiero systems, a much simpler strategy was used: if

a substring looked like a URL, a hashtag, or a mention, then it was protected from tokenization and truecasing during preprocessing of both training and test data. This generally caused such substrings to be OOV, so the decoder tended to copy them to the output.

OOV translation. We refined our CP1 OOV translator to break ties more intelligently, preferring more common English words over less common ones (assuming both are translations of IL words that look equally similar to the OOV).

System combination. We discovered quite late the that NIST scoring system gives detailed segment-level feedback information. In particular, we could discover which of our submissions outperformed which other submissions according to Bleu, not only across the 10% of Set E reported by NIST, but on each segment as well. For example, our "do-nothing, submit Oromo source" (corpus Bleu: 11.29) still outperformed our best Oromo translation system (corpus Bleu: 13.21) on 100 of 485 segments.

We felt that this detailed feedback was not appropriate and did not correspond to any realistic scenario. To show how it could be exploited, we produced a new unconstrained-track submission in which we automatically selected translations, segment by segment, from three previous submissions, based on Bleu scores from the NIST interface. For the 10% of the segments on which NIST gave feedback, this strategy improved Bleu by 2 points on Oromo (13.21 to 15.28, CP3u5) and 3.5 points on Tigrinya (17.61 to 21.19, CP3u9). We can consider this a kind of "oracle" system output selector.

We also built an SVM classifier to decide which system's output to use, on a segment-by-segment basis. This classifier was trained on the 10% of the data for which NIST provided feedback, and applied to the remaining 90% of the Set E data. We trained this classifier on three Tigrinya systems, achieving 44% test set accuracy (versus 34% for always choosing the best system, since the systems were evenly matched). By contrast, we were not able to build a good classifier for three Oromo systems. We do not know how these system combinations performed on the full Set E.

At no point did we manually look at source or target translations from Set E. We only looked at the numerical data provided by the NIST submission site. This meant that we developed SVM feature sets blindly, only imagining what properties of MT inputs and outputs might let us automatically guess which system's outputs were better than another.

Numbers, dates, and quantities. Late in CP3, we developed initial expression identifiers and translators for Tigrinya and Oromo.

Native Informant (CP3, Tigrinya and Oromo): We continued to have NIs gloss word-by-word in our Chinese Room interface, so that we could understand better how the languages work. In total, we got 110 sentences translated by Tigrinya ILs, plus 49 sentences from monolinguals in the Chinese Room postedited by NIs. We obtained 111 Oromo sentence translations resulting from Chinese Room editing (by NI or monolinguals), postedited by the NI; plus, 57 sentences from monolinguals in the Chinese Room, not postedited.

Chinese Room. We put up our first Chinese Room interfaces, with expanded functionality (such as user-to-user transfer of word glosses). The Tigrinya interface allowed us to translate sentences into English, but the Oromo interface remained difficult. This was the first time we experienced such a dearth of resources, and difficulty of language, in the Chinese Room. If humans cannot translate under these impoverished circumstances, we felt it calls into question asking the machine to do so. Nevertheless, we managed to translate for ourselves approximately 50 sentences in the Chinese Room.

We happened to include some English sentences from Set 1 in the Chinese Room, which underscored our problem with overtranslating Oromo tokens that should be left alone. For example, when given the sentence "Yet those who fled the injustice are facing rejection for the last 20 to 25 years", the Chinese Room's fuzzy-matching best suggested gloss is "Wriggle hut palisade night at Justice tree hat tree sheath at year 20 8mm 25 incense." Should our MT system attempt the same, we would trade in a large number of 4-gram Bleu matches for not even a single 1-gram Bleu match.

Checkpoint 3 submission results:

IL5/Tig	Description	Bleu
CP2c4	Best CP2 submission	17.54
CP3u1	SBMT + v5 dict + v3 data + NMT	17.61
CP3c8	Hiero + v6 dict + v6 data, Amh→Tig	15.41
CP3c4	Hiero + v8 dict + v6 data, Amh→Tig + URL	**18.54**

IL6/Orm	Description	Bleu
CP2c7	Oromo source untouched	11.29
CP3u2	CRF DNT tagger (*the* otherwise)	9.33
CP3u4	v5 Heuristic DNT tagger (& *the*)	10.08
CP3u6	+ v6 data + v7 dict + Som→Orm	**13.21**

Lessons Learned.

- The Chinese Room underscored the difficulty of Oromo MT, given the scant resources. We had never encountered a resource situation so limited that humans could not translate in the Chinese Room.
- It was more difficult than expected to predict which foreign words should not be translated. Of course, had we been permitted to manually inspect Set E source material, we would have been able to solve this problem.

B. Critical Additional Features and Tools

See above sections.

C. Other Data

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

D. Data Pre- and Post-Processing

See above sections.

E. Native Informant Use

See above sections.

F. Remaining Challenges

Oromo resources were small and out-of-domain, and the test data from "the wild" contained many words that should not be translated. Automatically identifying these should be a high priority going forward—this is a challenge because OOVs might be morphological variants of known words (which should be translated) or proper names (which should be copied). We also produced MT components for numbers, dates, and quantities late in the evaluation, so we should develop universal components that work out of the box (or nearly so) on the first day.

IV. Situation Frames from Text

The primary team consisted of Nikolaos Malandrakis, Pavlos Papadopoulos, Anil Ramakrishna, Karan Singla, Victor Martinez, Dogan Can 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, system combinations of the "MLP-LSA", "CNN-GRU", "LEIDOS" and "OSC" models. We also submitted results from two secondary models, a lexicon-based baseline system and a hierarchical attention model. 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 to documents and then attempt to localize those types to the available locations, then the resulting frames (and assigned text segments) are passed though the status detection models to get need, relief and urgency. Most of our core models very similar to those used in the 2016 iteration of the task. The "MLP-LSA", "CNN-GRU" and "LEIDOS" models were all submitted in 2016, but for 2017 we repeated the entire hill-climbing for hyper-parameter selection and tuned for f-score instead of precision. The models were also trained on a combination of text and speech SF datasets, with hill-climbing again used to decide which subsets of the data to use in each case.

Type Detection

a) The LEIDOS 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 trained on the ReliefWeb corpus and the word embeddings were initialized using the, publicly available and general purpose, GloVe embeddings. The final layer is composed of 40 binary classifiers, each corresponding to one topic or disaster type. To apply to the LORELEI SF task we simply created a deterministic mapping from some ReliefWeb to some LORELEI categories.

b) The OSC model: is another compositional CNN-GRU, similar to the Leidos model, but this time trained on the OSC corpus. The generation of LORELEI labels is again accomplished via a deterministic mapping. It was trained on the OSC corpus and the word embeddings were initialized using GloVe embeddings.

c) The CNN-GRU model: is the Leidos model, re-trained specifically for LORELEI. It shares the same topology as the ReliefWeb model, however to accommodate usage with very limited amounts of data the components have much lower dimensionalities. The first stage of training was performed using the ReliefWeb corpus and GloVe embeddings for initialization, the second stage involves removing the final layer of binary classifiers and re-training the entire network using a combination of SF speech and text data.

d) The MLP-LSA model: is a multi-layered perceptron applied to LSA document vectors. The LSA transformation was learned using the OSC corpus, which was 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.

e) The lexicon based model (lexica_baseline): is a logistic regression classifier trained with TF-IDF features extracted aat the document level from the HA/DR lexicon. This model acts as a baseline and is meant to give us an idea of what kind of gains (or losses) we are achieving by using more complicated machine learning algorithms.

f) Hierarchical Attention Model (HATT): : Similar to the LEIDOS model this is a compositional model but takes into account the document structure. The model first composes all words in each sentence using a word-level LSTM and attention layer to get sentence representations and then composes all sentence representations to a document representation using sentence-level LSTM and attention layer. The model essentially learns the weights for each word and each sentence in making a decision. We then replace the prediction layer with a new initiated prediction layer and then re-train the entire network again end-to-end on LDC-CMN dataset. As attention gives the importance of each word/sentence for predicting SF type, we plan to use this model to solve localization in the future.

Type Model Combinations

We used three combinations of the above models, achieved via max posterior decision-level fusion (similar to the union).

- "SYSCOMB" = "MLP-LSA" ∪ "LEIDOS".

- "SYSCOMB1" = "MLP-LSA" ∪ "CNN-GRU".
- "OPTIMIST" = "MLP-LSA" ∪ "CNN-GRU" ∪ "LEIDOS".

The system combinations are meant to increase the robustness of the output, e.g., we expect "MLP-LSA" to be the best preforming model for poor MT and "CNN-GRU" to perform best for high BLEU MT, therefore the "SYSCOMB1" should be more effective than either constituent model at leveraging MT of variable quality, even if not absolutely the best at any condition.

Localization

Most of 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 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 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.

Status Models

g) The ELISA-STATUS model: is composed of three independent SVM models for urgency, need and relief. Each model was trained on human-annotated English SF corpus, machine translations of the Uyghur unsequestered set, and LDC-CMN dataset. For each language, we extracted both lexical and affective features. Lexical features included document-level posterior probabilities for situation frames[1], and 500-dimensional LSA embeddings[2]. Affective features were extracted from the following sources: NRC Sentiment-Emotion lexicon and Emotiword. Affective features were centered, scaled, and reduced (5 component PCA). Lexical features were then concatenated and a second-pass dimensionality reduction (65 component PCA) was applied. Parameters were selected using a leave-one-language-out validation scheme.

h) The COLUMBIA Urgency model: is the model developed by the Columbia team which they graciously shared with us. For checkpoint 1 they labeled the English parallel data with emotion and urgency labels using a pre-trained emotion system. The tags were projected to IL and classifier trained in IL. Pre-trained emotion system: The parallel data was tagged using LSTM Recurrent Neural Net trained with a multi-genre English corpus (genres: emotional blog posts, tweets, news title, movie reviews). For checkpoint 2, used new sentiment and emotion labels on Set 0 parallel corpora, to train an urgency classifier and re-trained using self-training with urgency labels on Set 1. Sentiment was used to change urgency labels to false when sentiment=positive. Parallel corpora includes REFLEX for IL5 and sentiment labels were predicted on IL side of parallel corpora. Method for emotion: used the same LSTM Recurrent Neural Net to tag REFLEX parallel English data, and English part of set S Method for sentiment: used the same LSTM that was trained on English Twitter data.

[1]Obtained using SYSCOMB ELISA S.F. System

[2]Previously learned from ReliefWeb

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 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
- R libraries: xgboost
- Matlab

C. Native informant use

The initial intent was to use the native informant to annotate a few documents and use their input as part of reinforcement learning. That idea was abandoned due to time constraints and reliability concerns: training an annotator to perform the SF task requires many hours and we still would not be confident in the annotations provided. Instead we devoted our allotted time to improving the machine translation as it pertains to the detection of situation frames. We used the Leidos and OSC corpora to select English terms relevant to the task, mostly bigrams with a few trigrams, and had the native informants translate them to IL5 and IL6. The term list was selected using a combination of class-relevance and document frequency of ngrams followed by a round of manual post-hoc filtering. In total we had 20 NI sessions each spanning 1 hour (10 for each IL). In all sessions the task was the same: translate salient n-grams from English to IL to help with MT. We worked with two informants: *Native Informant 3* for IL5 and *Native Informant 6* for IL6. In total we were able to translate around 700 ngrams in IL5 and 975 ngrams in IL6.

Both the native informants in year two appeared considerably less trained compared to the previous year during the first checkpoint. As a result the number of translated ngrams was lower than we had hoped for from checkpoint 1.

a) IL5 CP1 meeting, 1 hour: NI3 was easy to communicate with and started the task without much confusion. However, he seemed distracted in some parts of the session which included a personal phone call several minutes long. He

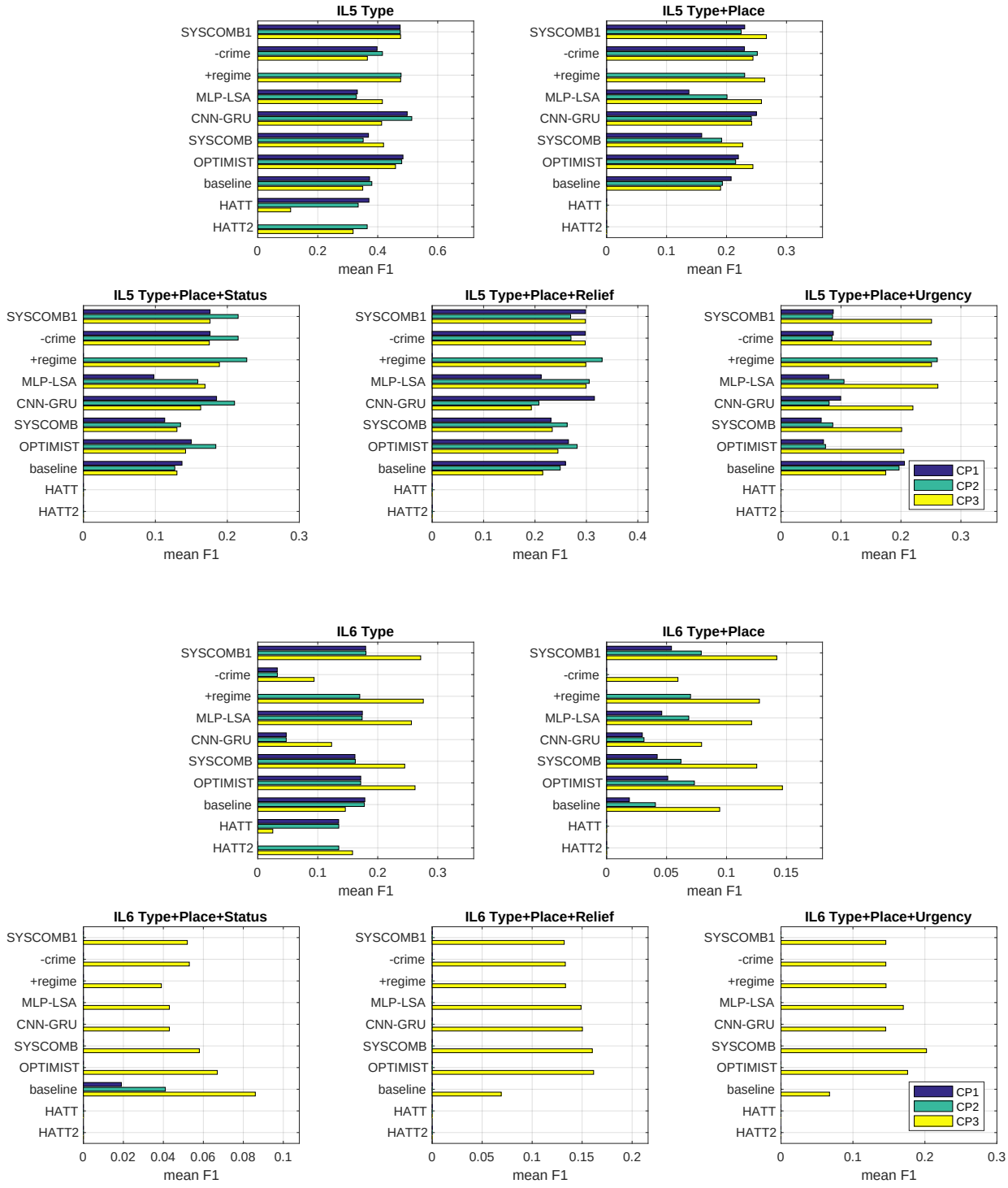

Fig. 4. Text Situation Frames reported mean F1 for all constrained submissions across all checkpoints.

also appeared to be delayed by his text entry interface while typing the translated words in IL5 script. Upon further request he stayed on a little longer (equal to the duration of his phone call) after the alloted one hour to help with the task. In total we obtained translations for 46 ngrams in the alloted time.

b) IL6 CP1 meeting, 1 hour: NI6 was relatively faster, in part thanks to the fact that IL6 uses Latin script. We were able to collect 81 n-grams in one hour.

1) Checkpoint 2: Both the native informants appeared to be better prepared by the second checkpoint.

a) IL5 CP2 meetings, 4 hours: We obtained 305 new translations in checkpoint 2.

b) IL6 CP2 meetings, 4 hours: We obtained 437 new translations in checkpoint 2.

2) Checkpoint 3:

a) IL5 CP3 meetings, 5 hours: We obtained 361 new translations in checkpoint 3.

b) IL6 CP3 meetings, 5 hours: We obtained 458 new translations in checkpoint 3.

D. The evaluation

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 very few training and evaluation samples for most of the issue types, so performance on them was not expected to be good - assuming we could even get a reliable estimate. For "Terrorism or other extreme violence" we had some confidence, for "Elections, politics and Regime change" almost all the samples we had were from the speech SF datasets and similarly for "Civil Unrest and wide-spread crime". The choice for checkpoint 1 was to not generate any "Elections, politics and Regime change" frames and make submissions both with and without "Civil Unrest and wide-spread crime" frames.

Evaluation of the individual models was performed by applying them to the parallel data provided in Set0 and on machine translations of the same documents. The main goal was to evaluate robustness to any occurring MT artifacts. Our examination showed that the models produced much smaller outputs on IL6 than IL5 but otherwise seemed to perform as expected, so no changes were made based on this examination.

Overall we made 8 constrained submissions, the results shown in Fig. 4. For this checkpoint Urgency was produced by the COLUMBIA system (for all systems but one), while Need and Relief were produced by the ELISA-STATUS system. Our primary system, SYSCOMB1, was submitted (across checkpoints) in three variants

1) the default (and primary) produces "terrorism" and "crime" frames, but not "regime change" frames
2) SYSCOMB1-crime is the same system, but not allowed to produce "crime" frames
3) SYSCOMB1+regime is the same system but allowed to produce "regime change" frames (so it produces all Types)

All submissions for checkpoint 1 apart from "SYSCOMB1-crime" include "Civil Unrest and wide-spread crime" frames. The results are very different in IL5 and IL6. IL5 represents a best case scenario, where MT and IE are very good and the performance is in line with the best numbers we got on our development sets. Performance on IL6 is less impressive, in large part because of the lower performing MT and IE inputs: the biggest indicator is the relative performance of "MLP-LSA" and "CNN-GRU", where the less robust compositional model reigns supreme in IL5, but the more robust bag-of-words model does best in IL6. The system combinations worked well, with SYSCOMB1 doing a good job of leveraging the better qualities of its constituent models and being among our best submissions for both languages. Another interesting observation is that the baseline model achieved the best results for Urgency. That turned out to be an accident: it used the ELISA-STATUS system for Urgency rather than the COLUMBIA system. In the following checkpoints we would revisit that.

Lessons Learned:

- We should trust our models more than our own intuition: despite our worries the crime labels worked out. We expended on that in checkpoint 2.
- Our in-house Status variable model worked better than anticipated, therefore we would have to consider switching to it.

Checkpoint 2

Our main modification for checkpoint 2 involved the use of EDL linking information to improve localization performance. As detailed in section IV-A we localize by creating sub-documents for each entity. For checkpoint 1 the grouping of entity mentions into entities was done via token matching: if two mentions had matching strings and entity types they were grouped into the same entity. For checkpoint 2 we used the entity cluster IDs as the grouping criterion. To accommodate the change we also changed the localization algorithm to allow for more localized frames to be produced, since now the danger of over-producing was mitigated.

Between checkpoint 1 and 2 we found a small bug in the Status assignment code that affected Status and Relief assignment if Urgency was set to True. The bug was corrected for checkpoint 2.

We made one additional submission, using a revised HATT model with one extra fully connected layer in the output, though it made little difference.

Finally, the results of checkpoint 1 forced us to revisit our assumptions about label trust and Urgency. We had our concerns about the "crimeviolence" labels and they proved incorrect, so for checkpoint 2 we added a submission that also included the "regimechange" labels. We were also concerned about the Urgency performance achieved by the COLUMBIA system, so we wanted to try the ELISA-STATUS for more than the baseline model. For checkpoint 2 we added a submission addressing both concerns: a variant of SYSCOMB1 that was allowed the generate "regimechange" labels and used the ELISA-STATUS system for Urgency.

The results for checkpoint 2 are shown in Fig. 4. For this checkpoint Urgency was produced by the COLUMBIA system, except for the baseline and "SYSCOMB1+regime" models, while Need and Relief were produced by the ELISA-STATUS system. Overall the change in localization, in conjunction with the improvements in the MT and EDL inputs lead to improved performance (on average) at the

"Type+Place" layer and beyond, though there was little change in the "Type" results. The inclusion of "regimechange" labels made little to no difference, perhaps indicating that the samples tagged with "regimechange" are mostly not in the evaluation set. Finally the systems using ELISA-STATUS for Urgency vastly outperformed the alternative, indicating we should switch to it for all systems.

Lessons Learned:

- There is more information in the EDL product that we are not exploiting enough.
- Having more samples from the Types we are expected to produce would have been useful. We would not need to guess if we have the correct version or how to interpret the definitions.

Checkpoint 3

By this point we did not expect any major changes to the MT and EDL inputs, since they had both reached a relatively mature point. The main modification made to the systems involved hashtags. Our colleagues working on EDL observed that a lot of the hashtags they encountered contained information that would be relevant to the SF task and the hashtags were in English so they would be useful regardless of MT performance. For example, consider the hashtag "#oromorevolution"; if we could parse it into "# oromo revolution", then that would be information that the SF models could use. To that end we implemented a dictionary-based string splitter that would split a hashtag to a sequence of valid words (included in the dictionary). Since multiple splits were possible, the shortest (lowest number of tokens) was selected. Hashtags could include names, like "oromo" in the above example, so to enable parsing we used the EDL linking information: we added all token appearing in the linked wikipedia page titles to the dictionary, so the splitter could handle entity names. The resulting hashtag splitter improved performance very little on our development sets, but we expected it to contribute more in cases of low MT performance, such as IL6. This splitter was used for all of the main models: everything apart from the baseline and the hierarchical attention model.

The other change was a global shift from the COLUMBIA to the ELISA-STATUS system for Urgency. All our submissions for checkpoint 3 used the ELISA-STATUS system.

The results for checkpoint 3 are shown in Fig. 4. For IL5 Type we got some mixed results, with most submissions dropping slightly in performance or staying the same compared to checkpoint 2. This can be attributed to the "CNN-GRU" model which dropped substantially in performance for checkpoint 3, for unclear reasons. For IL6 the hashtag parser lead to dramatic performance improvements of around 0.1 f-score at the Type level and the improvement cascades down to the other layers. The hashtag parser presumably did not have this much of an effect on IL5 because of the much better MT performance on that language. Finally, the change in Urgency lead to universally better results, more than doubling the performance of all models affected.

Lessons Learned:

- It is worth re-iterating that we should trust our models more.
- Perhaps we need to pay more attention to social media. Our efforts so far have mostly focused on the more complicated (and interesting) longer documents, but minor changes in twitter handling can have dramatic effects on overall performance.

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.
- We should take a closer look at social media, which may have received less attention than the other document categories. The relatively simple addition of a tag/name parser gave us a very large performance boost.
- 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 time we can have an evaluation with all the data annotated from the start. In 2016 the evaluation results were very different from the feedback scores received during the evaluation. Time will tell if 2017 will be similar.

V. Situation Frames from Speech

To produce situation frames from speech we followed a similar approach with the one described in the previous sections for text documents. An overview of our system is presented in Fig. 5. 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. BUT Automatic Speech Recognition (ASR)

The BUT ASR system training was mainly based on exploiting the NI's. We follow the direction of the previous evaluations, and make use of the advanced text-based system as described in the previous sections. We tuned system parameters on a defined held-out set (based on the NI input) w.r.t. the Word Error Rate (WER).

1) IL5—Tigrinya:

a) Data description: The acoustic model was first (pre-)trained using the Amharic LRLP corpus (decoded using our Amharic ASR system [9], and transliterated to Tigrinya (see Sec. II-A). Training data provided by NI informants (see

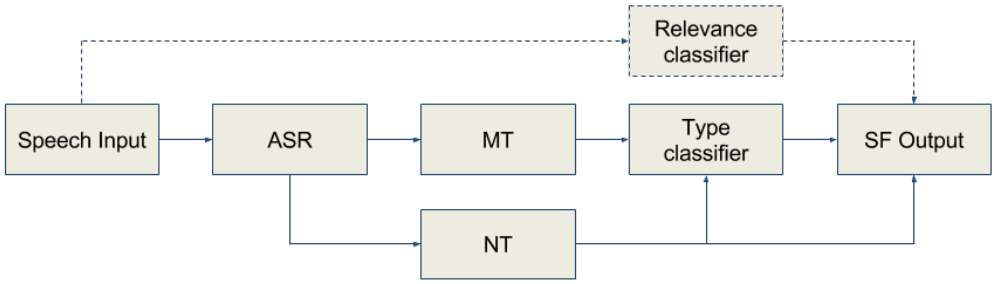

Fig. 5. Speech System Pipeline. Speech in the incident language goes through an Automatic Speech Recognition (ASR) component, whose output is utilized by the Machine Translation(MT) engine and the Name Tagger (NT). Once we have the translated output in English, as well as the place mentions, we identify the types of incidents occuring in the document and produce situation frames. Additionally a Relevance Classifier can be applied to the raw audio stream to influence the type confidence scores.

Sec. V-A3 below) were further used to MAP-adapt the system to Tigrinya. The data statistics are:

- NI data — 1455 utterances in 4.35 hours
- Amharic LRLP corpus — 10k utterances in 14.5 hours

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 [9] were used for the experiments. The major part of the training is similar to the procedure of November 2016 evaluations, as described in [10].

c) Acoustic model: The acoustic model is constructed with 269 graphemes. The graphemes act as syllables and hence position independent GMM-HMM models were trained. The models trained with G2P based phonemes did not perform better over grapheme systems. GMM-HMM models were used to extract alignments and 6 layer DNN with 1024 neurons were trained with RBM initialization. The DNN is trained with MultRDT features without any fMLLR transforms.

d) Language model: Unigram model was prepared from the expect-to-appear word-list, as described in Sec. II-F. 3 million utterances of Tigrinya text were used to build trigram model. The unigram and trigram models were interpolated with empirically chosen weights of 0.4 and 0.6, respectively. The lexicon contains 46k unique vocabulary list.

2) IL6—Oromo:

a) Data description: Training data were provided solely by NI informants (see Sec. V-A3 below). The corpus contains 1389 utterances in 4.95 hours.

b) Input features: The speech signal was pre-processed using multilingual+music VAD to remove music and non-speech portions. Multilingual-RDT (MultRDT) features [9] were used for the experiments. The major part of the training is similar to procedure in [10].

c) Acoustic model: The acoustic model is constructed with 26 graphemes (grapheme=phoneme). GMM-HMM models were used to extract the alignments, and 6 layer DNN with 1024 neurons was trained wih RBM initialization. The DNN is trained with MultRDT features without any fMLLR transforms.

d) Language model: Unigram model was prepared from the expect-to-appear word-list, as described in Sec. II-F. 3 million utterances of Oromo text were used to build a trigram model. The unigram and trigram models were interpolated with empirically chosen weights of 0.3 and 0.7, respecitvely. The lexicon contains 39k unique vocabulary list.

3) Native Informant for Speech: The strategy for us was to obtain as much training data for the acoustic models as possible. Following our November 2016 strategy, we split the NI sessions into reading aloud speech and speech transcription.

- Reading:
  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. The list of filtered sentences was then numbered and formatted into a googledoc. 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. As a backup, Appen was asked to record the sessions.
- Transcribing:
  In the transcribing scenario, we manually picked a set of short audio segments from the DEV (Set0) sets. We concentrated our effort on selecting the biggest variety of speakers and acoustic environments. We also made sure the segments were short enough for the NI to easily transcribe (max. 5 seconds).

This way, we conducted 6 reading and 4 transcribing sessions with IL5 NI's and 7 reading and 3 transcribing sessions with IL6 NI's.

B. UIUC Automatic Speech Recognition (ASR)

The UIUC/UW team used non-native human transcribers to generate a pseudo-phonetic transcription of the speech, which was time-aligned using ASR. Transfer learning from related languages (Amharic, Dinka, and Somali) gave us a simple IL G2P. Language models were generated from monolingual texts in the IL.

Non-native human transcription (mismatched crowdsourcing) was acquired from workers hired on Mechanical Turk. We split each IL .flac file into clips of about 1.25 seconds. For each clip, crowd workers on Mechanical Turk made 3 to 5 transcriptions. Workers were told to listen to each clip as if it were nonsense speech, and to write what it sounded like. They were told not to use complete English words to transcribe the non-English audio, since we've found that nonsense-word transcritions follow the phonetic content of an utterance better than English-word transcriptions. We verified the quality of the transcriptions, and rejected non-compliant workers.

A grapheme sequence in English text may represent a variety of phone sequences. By applying an English-language G2P to each space-bounded nonsense word, it is possible to generate a list of the different possible phone sequences that the transcriber might have intended to represent. An ASR trained using these audio clips, with these transcriptions, was then used to select the phone sequence best matching the audio. We trained two of these nonsense-English ASRs: one for IL5 audio, one for IL6 audio. The output of this process is a set of 3-5 candidate phone transcriptions for each audio clip (one for each crowd worker). The phone transcriptions were converted to single-chain FSTs, unioned to form a confusion network, and mapped from the English phone set into the phone sets of the ILs using our PTgen software (and using a cross-language phone map that we have previously published, and that is distributed on our github site).

An attempt was made to union crowd-worker phone transcriptions with the phone transcription produced by an Amharic ASR. The attempt failed, because of file-type incompatibility problems that we didn't have time to solve.

Phone transcriptions were converted to IL5 and IL6 word-level transcriptions using a dictionary based on monolingual text data. All space-bounded strings in the text data were treated as candidate words. Words containing non-native characters (non-Ge'ez for IL5, non-Latin for IL6) were excluded. Pronunciations of all remaining words were computed using G2Ps transferred from closely related languages. G2Ps in both cases were simple symbol tables (FSTs with no more than 100 arcs each). Symbol tables for both Tigrinya and Oromo are available in the LanguageNet, but were excluded from this exercise in order to avoid using any in-language resources. Instead, an Amharic symbol table was used to compute pronunciations in IL5, and the union of Dinka and Somali symbol tables was used to compute pronunciations in IL6. This process resulted in a pronlex of about 300k entries per language.

The pronlex for each language was converted to a trie, and searched for matches to any path through the transcription confusion network. We attempted a minimum string-edit-distance search, but found it too computationally expensive. The first set of generated transcriptions therefore used exact-match search, but contained very few words longer than about one syllable. In order to generate transcriptions with better length statistics, a limited sort of soft-matching was computed by clustering phones using a soundex-style clustering process, then performing an exact match on the soundex codes, resulting in the second set of generated transcriptions.

The second set contained many words that were OOV to the MT, especially in IL6, because orthography in both languages is so variable. For the third set of transcriptions, each output word was compared to a list of MT unigrams (IV words). If a word was OOV, we searched for IV words with the same pronunciation; if found, the IV word was substituted for the OOV.

The third set of generated transcriptions contained few place names. For the fourth set of generated transcriptions, each pronlex was enhanced using a long list of candidate spellings of IL5 and IL6 placenames, generated by RPI at an earlier stage of the competition.

C. Core algorithmic approach

Type Classification

The models that handle type classification in this task are similar with the ones in the text task. However, blind application of the type classifiers optimized for the text task yield inferior results in the speech task. We believe this is attributed to the noise introduced by the ASR component. Hence, the models need to adapt to this kind of noise. Hence, we built three different versions of each model based on different training sets.

In the first version, the system was trained on the representative Mandarin, Uyghur, and English text SF datasets, as well as a second set transcribed and translated speech SF sets for Turkish, Uzbek, Mandarin, and Russian. Although, we were not provided reference transcripts (thus cannot provide WER) we are confident that the respective ASR systems provide reliable transcriptions. In the second version we augmented the training set with Amharic transcriptions. Although the ASR system for Amharic did not produce transcripts of the same quality as the ones for the previous languages we believe that the similarity with the incident languages can enhance the performance. Finally, in the third version we included the Uyghur transcripts from the speech pilot. We expect that the ASR robustness for Uyghur will be comparable to the ones built for the two incident languages, thus adapting the models to the actual testing conditions.

In the following paragraphs, we will present the models that we employed and highlight their differences with those used in the text task.

a) CNN-GRU model: The CNN-GRU model is a similar model to the one described in IV-A. However, model parameters are randomly initialized instead of using a pre-trained Leidos model for initialization. We found that this "pre-training" step was not beneficial. Moreover, this model was overfitting faster than its text counterpart. We suspect that this happens because of the noise introduced by the ASR. The output layer consists of eleven independent binary classifiers, and the network was trained using binary cross-entropy.

b) MLP-LSA model: The MLP-LSA model is also a similar model to the one presented in IV-A. The LSA transformation was learned using the OSC corpus. Model parameters

were randomly initialized. Again we observed that the model was overfitting faster than the one trained for the text task. The output layer consists of eleven independent binary classifiers, and the cost function employed to train the network was binary cross-entropy.

c) Model Combinations: For our final system, we fused the outputs of the CNN-GRU and MLP-LSA models. Fusion was achieved using a max probability strategy. Combination with text models or the Leidos model were not found beneficial. Moreover, we developed an additional form of system combination that operates on the Situation Frame level. This method was motivated by the need to combine the output of systems built using different ASRs. In this approach, once the individual systems have produced types and we transform them to situation frames, we take the union of the two SF outputs. If the two systems produced the same frame, we average the confidence scores. We hope that using this approach can boost the performance at the localization level.

d) Model Variations: We have variations of the above models based on the data used to train them. We have found that the quality of the ASR component can influence the rest of the pipeline. To that end we training our systems using different sets to mirror those conditions.

e) Localization: Regarding localization, we follow a similar approach as the one described in IV-A. However, whenever a mention is detected we produce localized frames without any post-process filtering.

f) Relevance Classifier: We can optionally apply a relevance classifier in the audio stream. The relevance classifier is a SVM classifier using a 2nd degree polynomial kernel trained on low-level audio features to which we appended ivectors. The low-level acoustic features include various statistical functionals of speech properties (such as pitch, energy, and jitter) and were extracted across different languages to avoid capturing information specific to a particular language. Application of the relevance classifier alters the training procedure of the type classifiers. Since the goal of the type classifiers is to produce $p(type)$ and the the purpose of the relevance classifier to produce $p(relevance)$ we need to make the type classifiers generate conditional probabilities $p(type|relevance)$. Hence, when we apply the relevance classifier, the models are trained only with documents that contain at least one incident that would lead to a situation frame.

D. 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 transcribed and translated speech SF sets for Turkish, Uzbek, Mandarin, Amharic, Uyghur, and Russian were used to train models.

The main tools and software packages used were:

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

E. Native informant use

See Section V-A

F. The Evaluation

In this task we had just one checkpoint. We participated on the constrained scenario of both incident languages and in all three evaluation layers. We submitted multiple systems. The naming of the systems can be resolved as:

- Incident Language
- Which ASR system was used. If the naming include the string 'BUTUIUC' both the ASR systems were used to produce SFs. The SFs were combined using a union scheme with probability averaging.
- Application of Relevance classifier, e.g R means used NR not used. If the naming include the string 'RNR' both the ASR systems were used to produce SFs. The SFs were combined using a union scheme with probability averaging.
- Which datasets were used to train the type classifiers. Systems that include V3 in their names include the representative Mandarin, Uyghur and English text SF datasets as well as the transcribed and translated speech SF sets for Turkish, Uzbek, Mandarin, Amharic, Uyghur,and Russian. Systems with V4 are same with V3 except that Uyghur have been left out.
- Which Type classifier was used

Hence IL5_BUT_NR_V3_MLP refers to a system applied for incident language 5, using the ASR from BUT, without the relevant classifier, employing the MLP type classifier which was trained on the set described by V3.

For IL5 we submitted ten systems for the constrained scenarios, and participated in all 3 evaluation layers. The results based on the feedback provided by the organizers are presented in Table V-F.

TABLE XI
ELISA IL5 SF from Speech Results

System	Relevance	Type	TypePlace
IL5_BUT_R_V3_MLP	0.5048	0.2605	0.0118
IL5_BUT_R_V3_CNN_GRU	**0.5418**	0.2937	**0.0144**
IL5_BUT_R_V3_SYSCOMB1	0.5315	0.2833	0.0136
IL5_BUT_NR_V3_CNN_GRU	0.5376	0.3202	0.0132
IL5_BUT_NR_V3_SYSCOMB1	0.5395	**0.3321**	0.0136
IL5_UIUC_NR_V3_CNN_GRU	0.3268	0.0421	0.0001
IL5_BUT_NR_V4_SYSCOMB1	0.5129	0.3070	0.0122
IL5_BUT_NR_V4_CNN_GRU	0.5104	0.2946	0.0119
IL5_BUT_RNR_V3_SYSCOMB1	0.5576	0.3253	0.0139
IL5_BUT_NR_V3_MLP	0.4698	0.2603	0.0112

We observe that for the first evaluation layer (Relevance) a system using the relevance classifier provides the best performance. However, the for the second layer (Type) the system does not use the relevance classifier. Moreover, the type classifiers based on CNN-GRU provide better results than

the MLP ones. This indicates the ASR outputs have reliable outputs, since recurrent models fail when the input is poor, and the MLP models (which can be considered as Bag Of Words models) outperform the rest. For Type classification the best performing system is the combination of CNN-GRU and MLP (named SYSCOMB1). Finally, the performance on the third layer (TypePlace) is underwhelming. This was observed in the Speech Pilot task (Mandarin and Uyghur). We believe that the way this task is organized in inherently problematic. For example, in IL6 "Ethiopia" has about 244 different spellings. Thus, it is very unlikely a name detected by our system to match exactly the reference on mention string level. To solve this problem, in the short term (for this evaluation), you could collect all localization results from system output and reference, and create equivalence classes. This should not be difficult to do. We should integrate EDL into speech SF localization, so we can evaluate localization based on KB IDs/English translations, or simply use entity linking and clustering scores.

For IL6 we submitted ten systems for the constrained scenarios, and participated in all 3 evaluation layers. The results based on the feedback provided by the organizers are presented in Table V-F.

TABLE XII
ELISA IL6 SF from Speech Results

System	Relevance	Type	TypePlace
IL6_BUT_NR_V3_MLP	0.7166	0.2780	**0.0214**
IL6_BUT_NR_V3_CNN	0.6852	0.1953	0.0149
IL6_BUT_NR_V3_SYSCOMB1	0.6879	0.2266	0.0164
IL6_UIUC_NR_V3_MLP	0.6252	0.1160	0.0046
IL6_BUTUIUC_NR_V3_MLP	0.7062	0.2655	0.0098
IL6_BUT_R_V3_MLP	0.7409	0.2402	0.0152
IL6_BUT_R_V3_SYSCOMB1	0.7205	0.2039	0.0132
IL6_BUT_NR_V4_MLP	0.7188	**0.2810**	0.0212
IL6_BUT_R_V4_MLP	**0.7412**	0.2368	0.0151
IL6_BUT_NR_V4_SYSCOMB1	0.7039	0.2504	0.0174

Similar to IL5 a system using the relevance classifier provides the best performance for the first evaluation layer, and a system without it is outperforming the others in terms of the second layer (Type). Furthermore, we notice that in this case the MLP models give the best performance. Combination of MLP and CNN-GRU models actually hurts the system. This indicates the ASR outputs are not very reliable, and our system's decision are based on detected words. For the third layer (TypePlace) the results are similar to IL5.

Lessons Learned

- ASR performance is crucial to this task. If ASR output is poor the errors propagate to the rest of the pipeline.
- Data gathered from the speech task can help the training of models in the text task.
- Having some amount of annotated data (reference transcripts) for building a "reliable" ASR can help boost system confidence of the estimated types as well as improve localization information.
- The localization needs to be reorganized to something more meaningful.

G. Remaining Challenges

- The pipeline we employ is "fragile". Errors in a component propagate throughout the pipeline and hurt performance.
- 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. However, if ASR output is not satisfactory simpler models work best.

Acknowledgement

We would like to thank other ELISA team members who contributed to resource construction: Dian Yu and Samia Kazemi (RPI), and Chris Callison-Burch (UPenn). 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] 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.

[2] 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.

[3] H. Ji, "Mining name translations from comparable corpora by creating bilingual information networks," in *Proc. ACL-IJCNLP 2009 workshop on Building and Using Comparable Corpora (BUCC 2009): from parallel to non-parallel corpora*, 2009.

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

[5] F. Braune and A. Fraser, "Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora," in *Proc. COLING*, 2010, pp. 81–89.

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

[7] 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

[8] U. Hermjakob, K. Knight, and H. Daume, "Name translation in statistical machine translation: Learning when to transliterate," in *Proc. ACL*, 2008.

[9] 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.

[10] 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.