Mach Translat (2018) 32:31–43
https://doi.org/10.1007/s10590-017-9208-0

Combining rule-based and statistical mechanisms for low-resource named entity recognition

Ryan Gabbard[1] · **Jay DeYoung**[1] · **Constantine Lignos**[1] · **Marjorie Freedman**[1] · **Ralph Weischedel**[1]

Received: 31 May 2017 / Accepted: 7 October 2017 / Published online: 20 December 2017

Abstract We describe a multifaceted approach to named entity recognition that can be deployed with minimal data resources and a handful of hours of non-expert annotation. We describe how this approach was applied in the 2016 LoReHLT evaluation and demonstrate that both statistical and rule-based approaches contribute to our performance. We also demonstrate across many languages the value of selecting the sentences to be annotated when training on small amounts of data.

Keywords Named entity recognition · Low-resource NLP · Annotation

All work described in this article was performed at Raytheon BBN Technologies. Authors Freedman, Gabbard, Lignos, and Weischedel are currently affiliated with the University of Southern California Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, 90292, USA.

✉ Ryan Gabbard
gabbard@isi.edu

Jay DeYoung
jay.deyoung@raytheon.com

Constantine Lignos
lignos@isi.edu

Marjorie Freedman
mrf@isi.edu

Ralph Weischedel
weisched@isi.edu

[1] Raytheon BBN Technologies, 10 Moulton St., Cambridge, MA 02138, USA

1 Introduction

Named entity recognition (NER) in high resource languages has been the focus of many research programs, beginning with the MUC-6 Named Entity Challenge (Sundheim 1995) and continuing with the ACE, CoNLL (Tjong Kim Sang and De Meulder 2003), BOLT, GALE, and DEFT programs. The DARPA LORELEI (Low Resource Languages for Emergent Incidents) program seeks to expand this task into more languages while requiring systems to operate with fewer resources.

As a part of the LORELEI program, in 2016 NIST organized a community evaluation focused on low-resource human language technology in a surprise language for a disaster-focused scenario. A detailed description of the evaluation process, resources, and rules is contained elsewhere in this issue. We summarize the essential elements of the evaluation process below.

The language, Uyghur, and scenario, an earthquake, were announced at the beginning of the evaluation period. Participants were provided with data resources produced by the Linguistic Data Consortium (2016) and were evaluated at three checkpoints: 1 week after the surprise language announcement (Checkpoint 1, C1), 2 weeks (C2), and 4 weeks (C3). The data resources were also divided by checkpoint, with Set 0, consisting of a variety of resources, being made available immediately and additional sets of unannotated text (Sets 1 and 2) being made available after C1 and C2, respectively. Additionally, Set S, which consisted of unannotated English documents selected to be incident-relevant, was made available after C1.

Prior to C1, participants were provided 1 h of access to a native speaker of Uyghur, referred to as a native informant (NI). Prior to C2, four additional hours of NI time were available.

The evaluation was divided into two tracks. In the constrained track, only the data provided by the LDC and obtained from the NI could be used. The unconstrained track lacked this restriction. All work described in this paper was submitted as a part of the constrained track. At each checkpoint, the system was evaluated on its ability of the system to identify names of the following four types: geopolitical entities (GPE), locations (LOC), organizations (ORG), and persons (PER).

In the following sections we will describe our participation in this evaluation with respect to named entity recognition. Additionally, in Sect. 4 we will describe experiments performed after the evaluation to judge how certain aspects of the evaluation system would adapt to a variety of other languages.

Our system combines aspects of rule-based and statistical systems for NER. Briefly, relevant prior work includes rule based name-finding in low resource languages, e.g., Urdu (Riaz 2010); supervised models with plentiful data, e.g., CRFs for Hindi on the TIDES program (Li and McCallum 2003); unsupervised name list generation (Collins and Singer 1999; Nadeau et al. 2006); and more recently multilingual neural NER (Lample et al. 2016; Bonadiman et al. 2015).

Also relevant to our work are approaches for domain adaptation (Sun et al. 2016) and active learning (Settles 2010) for extending NER approaches. Notably similar in concept to our approach for automatic annotation selection is Zhang et al. (2016), in which the authors use active learning to train a CRF and automatically extracted rules to annotate documents.

Table 1 Data resources used during the evaluation

Resource	Size	C1	C2	C3
Unannotated monolingual text (tokens)	32.2 M	✓	✓	✓
Grammar book sentence annotation (tokens/sentences)	697/318	✓		
Grammar book name lists (names)	70	✓	✓	✓
Parallel dictionary (names)	330	✓	✓	✓
Wikipedia list of local place names (names)	260	✓	✓	✓
NI sentence annotation (tokens/sentences)	4.3 k/168		✓	✓
GeoNames (names)	2.2 k	✓	✓	✓
NI elicited names (names)	90			✓
Chinese-Uyghur bilingual lexicon lists (names)	3.6k		✓	✓

2 Evaluation resources

Table 1 describes the resources we used for named entity recognition and summarizes each resource's approximate size and the evaluation checkpoints in which it was used. In some cases there were minor modifications to a resource between checkpoints (e.g., removing unreliable strings from lists; adding the Sets 1 and 2 material to the unannotated corpus).

2.1 Name lists

We semi-automatically extracted lists of names from several resources, summarized below.

Uyghur-English parallel dictionary[1] We used patterns (e.g., English-side casing) over the English lexicon entries to extract potential names. Name types were assigned by an English speaker who manually reviewed the English gloss and transliteration of any potential name that appeared in a 10 k document subset of Set 0. We augmented these lists by aligning pre-existing English name lists with the English glosses.

Xinjiang places list[2] We manually mined the Uyghur place names from the Wikipedia category link provided by LDC. We cleaned this list of place names by removing designator words (as identified using the lexicon).

Chinese-Uyghur dictionary[3] A Chinese speaker provided patterns for semi-automatically extracting names from the Chinese-Uyghur dictionary. These patterns were used to extract additional name lists.

English-language Uyghur grammar[4] We manually extracted names that appeared as examples in the parallel grammar.

[1] IL3_dictionary.xml; LDC-provided.

[2] xinjiang_places.pdf with link to Wikipedia; LDC-provided.

[3] link in CategoryII_list.pdf; LDC-provided.

[4] parallel_grammar.pdf; LDC-provided.

Table 2 Sentence selection heuristic limits

Restriction type	Limit
# Tokens	≤ 35
# Matched names	≤ 10
Fraction of tokens matched as names	≤ 0.5
# Commas	≤ 2

GeoNames Our system uses a snapshot of GeoNames (Wick 2016) downloaded in January 2016. We used the GeoNames feature codes to map GeoNames entries to GPE and LOC types.

Native informant During our first native informant session, we augmented our name lists by asking the native informant to list common names. During later native informant sessions, we asked the native informant to translate names from the Set 0 situation description and to approve system-hypothesized names.

2.2 Native informant interactions

We used the 1-hour time slot before C1 to ask the native informant to do the following:

(a) Provide examples of common Uyghur names (e.g., list five schools/universities, list five famous people).

(b) Perform in-context named entity labeling using a browser-based tool. The browser-based tool requires that the annotator select a name span using the mouse and use a key to label the text. Manual review of the in-context annotation suggested that the NI did not understand the initial version of the task and had difficulties selecting the correct span. We did not use the in-context annotation performed in this session in our submissions.

We used four 1 h blocks with the NI before C2. During each block we asked a small number of targeted questions (e.g., *Is "…" a named river?*) and then asked the NI to perform in-context annotation. In-context annotation was performed in a Google Sheets spreadsheet that had been pre-populated with system-assigned labels. For each token, the spreadsheet included the Arabic-script token, its transliteration, its gloss from the English-Uyghur dictionary, and a system-assigned NER label (if any). During the session we used the transliteration/translation and Google Sheets's collaborative editing to answer the NI's questions and identify annotation errors.

We semi-automatically selected high-yield sentences for in-context annotation using heuristics, including the presence of names from our name lists and the occurrence of other name indicators mined from the lexicon and grammar (e.g., person titles). We filtered the unannotated sentences by the limits described in Table 2, and then ordered this output by the number of found indicators. English speakers reviewed potential sentences (using their lexicon-derived glosses and transliteration) to ensure that the sentences were representative of typical, well-formed data. The bulk of our in-context annotation was performed over Set 0 sentences. Some annotation was performed over Set 1 sentences.

2.3 Grammar and lexicon

We used the English-language Uyghur grammar in two ways:

(a) Defining a transliteration of the Arabic script into Latin script. We used this as the primary internal processing character set for our system as well as for review of system output. This allowed English-speaking developers to recognize many Uyghur names when inspecting the data and allowed them to discuss tokens with the NI.

(b) Generating a database mapping Uyghur word forms to their stems and morphosyntactic properties. Because vowel harmony is frequent in Uyghur, rather than using unsupervised morphology we built this database using hand-written rules based on the English-language Uyghur grammar book applied to to the lexicon and our name lists. These rules accounted for most nominal morphology at C1 and C2 and for some verbal morphology as well at C3. Morphological information was used to match lists and portions of patterns against stem forms regardless of inflection, and as features in some sequence model submissions.

2.4 English speaker interactions

In addition to the development tasks, English speakers (developers and/or annotators) reviewed a summary of primary system output on a 10 k document subset of Set 0 before each checkpoint. Review typically consisted of:

(a) Identification of unrecognized names in the 50 most frequent names and subsequent in-context review of those names, using lexicon lookup and transliteration. At C3, the review of unrecognized names was extended to the 250 most frequent names. The post-processor removed names identified as false alarms by manual review.

(b) Scaning the full set of names for unexpected output (e.g. very long names; non-word character names). This helped to identify patterns to apply in post-processing (see Sect. 3.1).

3 Evaluation results

3.1 Configurations

The configurations submitted to the evaluation along with their scores can be found in Table 3. The configurations differed in the following ways:

Lists indicates the use of the name lists discussed in Sect. 2.1. At runtime any token sequence found on these lists was marked as a name of the appropriate type. If morphological analyses were available, then any known inflected form of a word on the list would match. These lists changed from checkpoint to checkpoint based on feedback from the NI, developer inspection of the unannotated corpus, etc.

Table 3 Performance of evaluation systems

	Lists	Morph	Pats	CRF	Clust	UMorph	MSP	Post	Ckpt	P	R	F
A	✓							✓	C1	46.5	17.8	25.8
B	✓							✓	C2	49.0	21.0	29.4
C	✓							✓	C3	49.8	21.1	29.6
D	✓	N						✓	C1	56.0	33.0	41.6
E	✓	N						✓	C2	60.4	40.2	48.3
F	✓	N+V						✓	C3	61.4	40.4	48.7
G	✓	**N**	✓					✓	**C1**	**56.6**	**34.3**	**42.7**
H	✓	N	✓					✓	C2	60.8	43.0	50.4
I	✓	N+V	✓					✓	C3	61.7	43.1	50.8
J		N		✓	01SE	✓			C2	60.4	43.5	50.6
K		N+V		✓	01SE		✓		C2	60.3	42.2	49.7
L	✓	N	✓	✓	0SE			✓	C1	48.0	36.5	41.4
M	✓	N	✓	✓	01SE			✓	C2	60.6	55.3	57.8
N	✓	N	✓	✓	01SE	✓		− Cons.	C2	60.4	55.0	57.5
O	✓	**N**	✓	✓	**01SE**	✓		✓	**C2**	**60.3**	**55.1**	**57.6**
P	✓	N+V	✓	✓	01SE			✓	C3	61.8	55.1	58.3
Q	✓	**N+V**	✓	✓	**01SE**		✓	✓	**C3**	**61.9**	**55.0**	**58.2**
R	✓	N	✓	✓	01SE		✓	✓	C3	61.6	54.8	58.0
S	✓	N+V	✓	✓			✓	✓	C3	60.3	47.0	52.8
T	✓	N+V	✓	✓	01SE	✓	✓	✓	C3	61.7	54.9	58.1
U	✓	N+V	✓	✓	012SE		✓	✓	C3	61.5	54.5	57.8
V	✓	N+V	✓	✓	01SE		✓	+ Retok	C3	61.9	55.1	58.3

Bold systems were the primary submission for each evaluation checkpoint. For a key to the columns, see Sect. 3.1 Precision, recall, and F1 are provided for each of the three checkpoints

Morph indicates the use of our handwritten morphological analysis rules described in Sect. 2.3. Support for verbal morphology was added at C3.

Pats indicates the use of handwritten patterns and post-processing developed by an English speaker using the lexicon and targeted questions for the native informant. These rules were not wholly independent of the CRF model since some target observed sequence model mistakes. The patterns included at each checkpoint were as follows:

1. Expand found LOCs to include any preceding directional modifiers (C1).
2. If a designator word is seen, search backwards for a GPE or LOC, allowing only certain words to intervene. If one is found, expand it to include all text up to and including the designator (C1).
3. Identify schools and universities (C1).
4. Identify person names of the form *title X punctuation* or *title X Y* where *X* and *Y* are either unknown or appear on a list of unambiguous names.
5. Delete names which are all puncutation (C1).

6. If the Uyghur word for *river* is seen, make a LOC containing that token, the preceding word, and the following word if it is the word for *valley* (C2).
7. Identify newspapers (C2).
8. If there are two adjacent tokens, one of which is on a list of ‘risky’ name words and the other of which is on a list of ‘certain’ name words, mark the two tokens as a PER (C2).
9. When certain designators are seen, combine them with the immediately preceding token to make a GPE (C2).
10. Identify organization names (C2).

No new patterns were added for C3.

CRF indicates the use of a conditional random field (Lafferty et al. 2001) NER model using a BIO encoding (Ramshaw and Marcus 1999). At C1, this system was trained on sentences extracted from the English-language Uyghur grammar. An English speaker used the English glosses of example sentences to perform annotation. At C2 and C3, the training data was replaced with sentences annotated by the NI.

L1 regularization was applied with a coefficient of 0.1 and the model was optimized using AdaGrad (Duchi et al. 2011) with a learning rate of 0.1, $\epsilon = 10^{-6}$,[5] and minibatch size 64. Training was terminated when the loss had failed to improve over three iterations. Hyperparameters were selected based on previous work with Turkish and Uzbek.

The feature set used was:

1. Token features for the focus token, the preceding token, and the following token:
   (a) the token itself
   (b) whether the token is all-caps
   (c) whether the token is capitalized
   (d) whether the token is all digits
   (e) whether the token is alphanumeric
   (f) whether the token is all whitespace
   (g) whether the token appears to be a URL
   (h) whether the token is a currency symbol
   (i) whether the token is all punctuation and, if so, whether it is multiple codepoints
2. Sub-token features for the focus token only:
   (a) portions of the token separated by hyphen
   (b) the word shape of the token[6]
   (c) each individual character of the token
   (d) prefixes and suffixes of length 1–4

[5] This is the numerical stability parameter typically used in AdaGrad implementations.

[6] The word shape feature collapsed all consecutive letters in a name to a single letter to attempt to identify punctuation patterns. For example, the name *Bob* would have the shape *a*, while @*Bob* would have the shape @*a*.

All emissions features were conjoined with both the BIO label variable and a backed-off version with only the entity type. The features were not tuned for Uyghur in particular.

Clust indicates the use the Brown clusters of the focus, previous, and next token truncated to 8, 12, 16, and 20 bits as features in the sequence model. The field contents shows which of Sets 0, 1, 2, S, and E were used for inducing Brown clusters. Note that except in one variant Set 2 was always excluded from Brown clusters (see below).

UMorph indicates the use in the sequence model of features derived from unsupervised morphological analyses provided by the University of Pennsylvania (Xu et al. 2017).

MSP indicates the use in the sequence model of features derived from our handwritten morphological analyses, such as de-inflected forms and morphosyntactic properties.

Post indicates that post-processing rules were applied. These were:

1. Rule-based identification of Twitter handles. These were assigned type PER unless they appeared on a list of ORG handles manually collected from Sets 0, 1, and 2 (e.g., *@YouTube*) (C1).
2. All names were trimmed of "bad tokens" (C2). A token was bad if it matched regular expressions for file names, URLs, keyboard shortcut sequences, etc. We deleted any names which, after trimming, met any of the following conditions: only one character long, on a list of known bad names, whose transliteration lacked any ASCII letters, or which starts with a lowercase ASCII character in the untransliterated text. The known bad names were identified by manual English speaker review of high frequency names in Set 0.
3. Any name which appeared to be an e-mail address was changed to have type PER (C2).
4. Any name which appeared on a list of known locations was changed to have type LOC (C2).
5. Require that any multi-token name found in one place in a document would be marked in all its occurrences in the document (C2). −*Cons* indicates a C2 or later configuration with this rule disabled.

   The post-processing rules for C3 were essentially the same with more items added to various lists based on manual corpus inspection. *Retok* indicates using an alternate tokenization from that provided by the LDC. We observed in many cases the tokenization of the data provided by the LDC was problematic with respect to punctuation. In this variant we wrote our own tokenizer which was applied to all training and runtime data, modifying the LDC's tokenization. For the final output, we projected the name boundaries we found back to the LDC's tokenization, since the evaluation annotation was performed with respect to that tokenization.

3.2 Discussion

Evaluation scores for our submitted configurations can be found in Table 3. During the evaluation we selected which configurations to use as our primary submissions by tracking performance on the NI's annotation using cross-validation.

Our simplest baseline submissions (A, B, C) simply matched name lists derived from various sources (Table 1). The name lists were edited by developers across the checkpoints (adding names noticed during inspection of the corpus and removing names noticed to be erroneous). Names projected from the Chinese-Uyghur dictionary were added at C2. Names elicited from the NI were added at C3. The performance of this baseline increased sharply from C1 to C2 (25.8–29.4) but only very slightly from C2 to C3 (to 29.6). We attribute this to a combination of the Chinese-derived name lists and significant improvements to the patterns and post-processor at C2.

A stronger baseline can be obtained by augmenting name list matching with our handwritten morphological analyses (D, E, F). This provides almost a 20 point boost in recall. It also magnifies the boost observed at C2 (A to B improves recall by 3.2 but D to E improves recall by 7.2). Overall, morphological analyses boosted C3 F1 from 29.6 to 48.7.

Adding handwritten patterns to this baseline (G, H, I) produced a modest gain (F to I improves F1 from 48.7 to 50.8). The magnitude of this gain increases at C2, probably due to the expanded C2 pattern set.

Adding a CRF-based trained model (L, M, Q) generally improves performance significantly (e.g. I to Q improves F1 from 50.8 to 58.2). The exception is at C1 (L) where the only available training data was grammar book glosses, resulting in poor performance for the CRF. Contrastive submissions which use the sequence model without lists or rule-based patterns and post-processing (J, K) were made at C2 and C3. These show significant performance drops (Q to K drops F1 from 58.2 to 49.7), indicating that the sequence models and hand-tuned rules both make significant independent contributions.

Configuration S, which removes Brown clusters from our primary C3 configuration Q, suggests that Brown clusters account for much of the value of the CRF model, dropping performance from Q's 58.2 to 52.8. Cross-validation experiments had suggested that adding Set 2 data to the Brown clustering input was slightly harmful. This was confirmed by a constrastive submission to Checkpoint 3 (configuration U) which included Set 2 data lagging the otherwise identical primary configuration Q by 0.4 points F1.

The impacts of unsupervised morphology (T), rule-based morphology features (P,) the document consistency rule (N), and retokenization (V), and adding verbal morphology (R) were negligible.

4 Other languages

4.1 Data conditions

For the LORELEI program, the Linguistic Data Consortium provided NER annotation and general resources for many languages, including Amharic, Farsi, Hungarian, Rus-

Table 4 Performance of the SEQUENCE on the full data of each pack (75% used for training, 25% for testing)

Language	F1	# Training sentences (k)
Amharic	69.5	4.1
Farsi	57.7	3.0
Hungarian	60.1	3.0
Russian	68.8	6.5
Somali	80.0	2.6
Spanish	63.2	2.3
Turkish	75.2	4.1
Uzbek	76.4	8.2
Vietmanese	60.0	3.0

sian, Somali, Spanish, Turkish, Uzbek, and Vietnamese.[7] In the 2016 evaluation on Uyghur, our NI was able to annotate 168 sentences in 4 h. To better understand NER performance in low-data conditions, we experimented with reduced training data in these languages by selecting sentences for training using the technique from Sect. 2.2. We used GeoNames filtered by language to generate the name lists for selection.

25% of the training data was reserved for testing prior to performing sentence selection. The remaining 75% was divided into three portions. From each portion, we selected training set sizes ranging from 27 to 168 sentences to match the amount of NI training data we had acquired after each hour of NI annotation in the evaluation. We additionally added a 300 sentence training set for reference. These three distinct training sets for each language and data size combination allow us to examine the variability of the sentence selection process.

4.2 Models

For each language and at each training data size, we trained two models:

EXACT MATCH Memorizes the names seen in training and at runtime labels any instance of them with the type seen in the training data. It prefers matching longer sequences of tokens over shorter ones. If the same sequence of tokens is seen with multiple entity types in the training data, the first type seen is used.

SEQUENCE A CRF model using the language-general features described in Sect. 3.1 and optimized in the same way. Performance of SEQUENCE on the full data of each pack is given in Table 4.

[7] Arabic and Mandarin were also provided but we exclude them from our experiments here due to data processing issues. Yoruba is excluded because it had too little data for meaningful experiments. Hausa was excluded because the data did not annotate the GPE type. LDC catalog numbers were 2014E115, 2015E70, and 2016E{29,87,91,93,95,97,99,103}.

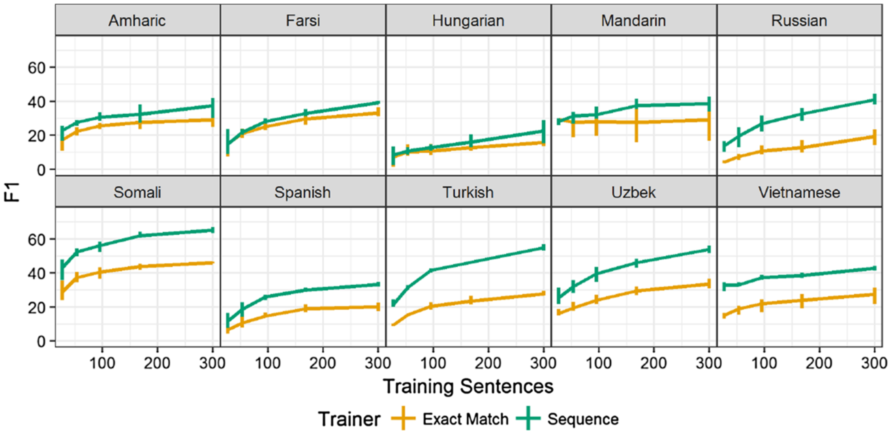

Fig. 1 Performance of EXACT MATCH compared to SEQUENCE for each language and training data size combination

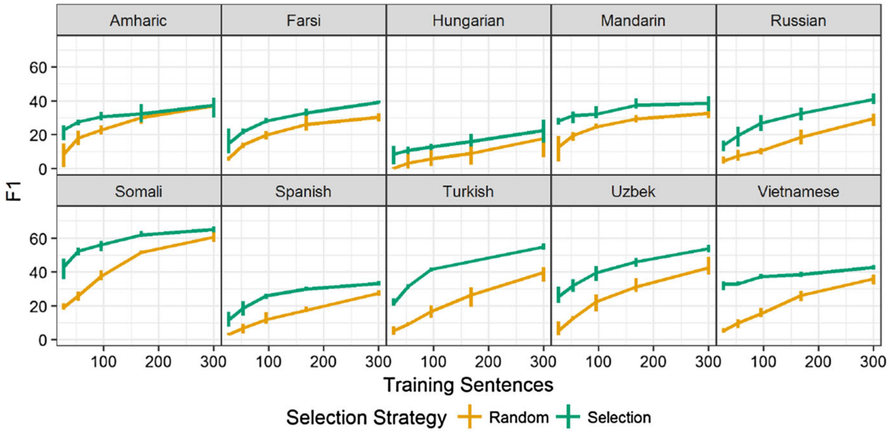

Fig. 2 Performance of informed and random training sentence selection for each language and training data size combination

4.3 Results

Figure 1 shows the performance of EXACT MATCH compared to SEQUENCE for each language and training data size combination. The vertical bars give the range of performance across the three partitions of the data for sentence selection. SEQUENCE matches or outperforms EXACT MATCH even at the 27 sentence data point and consistently maintains or grows its advantage as the amount of training data increases.

Figure 2 shows the performance of selecting training sentences using the strategy described above compared to selecting them randomly. Informed selection consistently outperforms random selection, sometimes by a wide margin, although the gap tends to shrink as the amount of training data grows. Note that in this experiment, informed selection has far less data to select from than was available in the 2016 evaluation described in Sect. 3.

Table 5 Variability of sentence selection performance across the three disjoint data partitions at the 168 sentence data point

Language	F1 Mean	Min F1 (%)	Max F1 %
Amharic	32.5	− 7.5	+ 8.7
Farsi	33.0	− 2.6	+ 3.6
Hungarian	16.0	− 3.7	+ 5.5
Russian	32.7	− 6.4	+ 4.8
Somali	62.0	− 3.6	+ 5.7
Spanish	30.1	− 1.9	+ 1.5
Turkish	46.5	− 1.0	+ 0.7
Uzbek	46.1	− 5.5	+ 4.1
Vietmanese	38.7	− 2.5	+ 2.1
Average		− 3.8	+ 4.3

Table 5 shows the variability of sentence selection performance across the three disjoint data partitions. The average variation in F1 between the extreme of the three runs and the mean is small: around 4% relative. The maximum is only 9% relative.

5 Conclusions

Our key observations are:

1. At this level of data resources, using trained and rule-based models together is significantly better than either alone. Much of the value of trained models seems to come from Brown clusters.
2. At least for Uyghur, morphological analysis has a very large impact on performance.
3. Tools which applied morphological analysis and bilingual lexicon lookup to provide English-speaking developers with a understanding of Uyghur text were vital for writing patterns and post-processors and for interacting with the NI.
4. Even simple techniques for informed selection of sentences for annotation can be very effective in low resource scenarios.

Acknowledgements This material is based upon work supported by the the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0113. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. (Approved for Public Release by DARPA on Aug 29, 2017 (DISTAR Approval #28392) , Distribution Unlimited)

References

Bonadiman D, Severyn A, Moschitti A (2015) Deep neural networks for named entity recognition in Italian. CLiC it 51–55

Collins M, Singer Y (1999) Unsupervised models for named entity classification. In: Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora, pp 100–110

Duchi JC, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:2121–2159

Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, ICML '01, pp 282–289

Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. CoRR abs/1603.01360, http://arxiv.org/abs/1603.01360

Li W, McCallum A (2003) Rapid development of hindi named entity recognition using conditional random fields and feature induction. In: ACM transactions on Asian language information processing, pp 290–294

Linguistic Data Consortium (2016) LORELEI IL3 incident language pack for year 1 Eval. LDC2016E57

Nadeau D, Turney PD, Matwin S (2006) Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In: Proceedings of the 19th international conference on advances in artificial intelligence: Canadian Society for Computational Studies of Intelligence, Springer, Berlin, Heidelberg, AI'06, pp 266–277

Ramshaw LA, Marcus MP (1999) Text chunking using transformation-based learning. In: Armstrong S, Church K, Isabelle P, Manzi S, Tzoukermann E, Yarowsky D (eds) Natural language processing using very large corpora. Springer, The Netherlands, Dordrecht, pp 157–176

Riaz K (2010) Rule-based named entity recognition in urdu. In: Proceedings of the 2010 named entities workshop, Association for computational linguistics, Stroudsburg, PA, NEWS '10, pp 126–135

Settles B (2010) Active learning literature survey. In: Computer sciences technical report, University of Wisconsin-Madison

Sun H, Grishman R, Wang Y (2016) Domain adaptation with active learning for named entity recognition. In: Sun X, Liu A, Chao HC, Bertino E (eds) Cloud computing and security: second international conference. Revised Selected Papers, Part II, Springer International Publishing, Cham, ICCCS 2016, Nanjing, China, 29–31 July 2016, pp 611–622

Sundheim BM (1995) Overview of results of the MUC-6 evaluation. In: Proceedings of the 6th conference on message understanding, Association for Computational Linguistics, Stroudsburg, PA, MUC-6 '95, pp 13–31

Tjong Kim Sang EF, De Meulder F (2003) Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003 - Vol4, Association for Computational Linguistics, Stroudsburg, PA, CoNLL '03, pp 142–147

Wick M (2016) Geonames ontology. http://www.geonames.org/about.html

Xu H, Marcus M, Ungar L, Yang C (2017) Unsupervised morphology learning with statistical paradigms, unpublished manuscript

Zhang B, Pan X, Wang T, Vaswani A, Ji H, Knight K, Marcu D (2016) Name tagging for low-resource incident languages based on expectation-driven learning. In: Proceedings of ACL 2016