Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6983–6987
Marseille, 11–16 May 2020

Dragonfly: Advances in Non-Speaker Annotation for Low Resource Languages

Cash Costello, Shelby Anderson, Caitlyn Bishop, James Mayfield, Paul McNamee
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road, Laurel, MD 20723 USA
{cash.costello, shelby.anderson, caitlyn.bishop, james.mayfield, paul.mcnamee}@jhuapl.edu

Abstract

Dragonfly is an open source software tool that supports annotation of text in a low resource language by non-speakers of the language. Using semantic and contextual information, non-speakers of a language familiar with the Latin script can produce high quality named entity annotations to support construction of a name tagger. We describe a procedure for annotating low resource languages using Dragonfly that others can use, which we developed based on our experience annotating data in more than ten languages. We also present performance comparisons between models trained on native speaker and non-speaker annotations.

Keywords: named entity recognition, non-speaker annotation, low resource languages

1. Introduction

Neural named entity recognition (NER) models require large amounts of training data to achieve state-of-the-art performance. A significant challenge with low resource languages is locating native speakers to perform these annotation tasks. To address this, we have created a tool for annotating documents in a language that the annotators do not speak. Our tool, Dragonfly, has been improved through our experiences annotating text in 13 different languages. It is available as open source software on GitHub.[1]

The development of this tool was motivated by the LoReHLT evaluations (Christianson et al., 2018), which included an evaluation of NER for surprise low resource languages with tight time constraints (approximately 1 to 2 weeks depending on the year). Three of the teams adopted non-speaker annotation as a key enabling technology for the evaluation, and developed annotation tools in parallel. TALEN (Mayhew and Roth, 2018) transforms the text towards English by transliterating non-Latin scripts and performing word replacement using a bilingual dictionary. The ELISA IE annotation tool is described in (Lin et al., 2018) along with an early version of Dragonfly. Both of these tools display the original text with transliterations and translations shown at the sentence level. This paper highlights the advances we have made to Dragonfly based on two additional evaluations. These advances make non-speaker annotation both faster and more accurate.

For each low resource language in the evaluation, monolingual text was provided from multiple sources (news, discussion forum/blogs, and Twitter). In addition, a bilingual lexicon, a grammar sketch, and a small amount of parallel text were included in the language pack. The minimum set of resources required for annotating with Dragonfly is the monolingual text in the language of interest and a small bilingual lexicon with some names of entities in it.

2. Requirements

When annotating the names of entities in an unfamiliar language, the annotator comes across two types of entities. The first of these are entities that are already known to the annotator. These are typically internationally-known entities, which include countries, major cities, and heads of state. Given some basic sentence context and a transliteration that allows the annotator to pronounce the words of the sentence, we have found that annotators can recognize many of these entities. The second type of entity are those that are local in context and unknown to the annotator. This includes the names of towns, villages, rivers, roads, chiefs of police, and local politicians. To discover these entities requires more local semantic and syntactic information, along with knowledge of the local area and people mentioned in the data set.

We see the following requirements, based on our prior work (Lin et al., 2018), as most critical for non-speaker annotation tools:

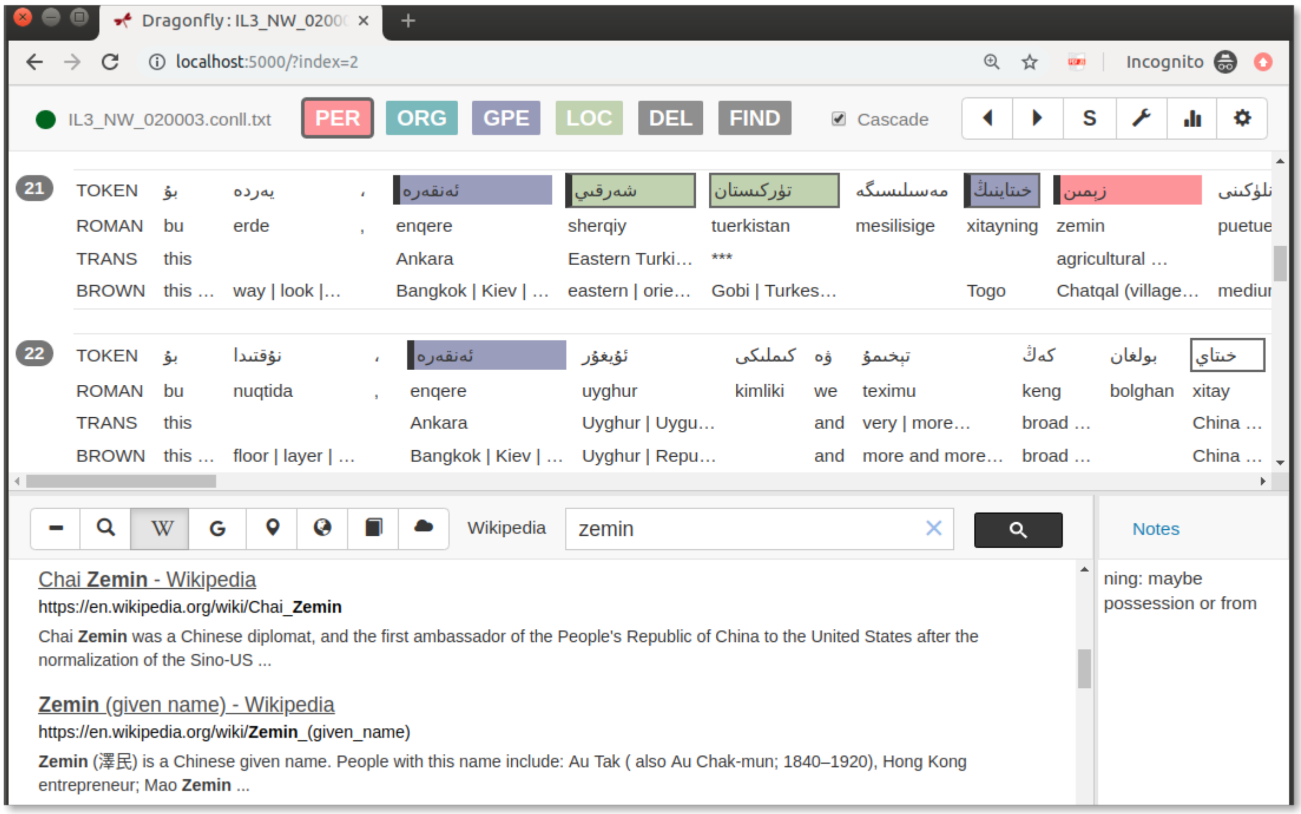
Word recognition. Presentation of text in a familiar alphabet, with identifiable word and sentence boundaries, makes it easier to see similarities and differences between text segments, to learn aspects of the target language morphology, and to remember previously-seen sequences.

Word pronunciation. Because names of entities outside of a language's geographic footprint enter a language through transliteration, being able to pronounce words is particularly important for annotating named entities (especially international entities). The pronunciations can be exposed either through a formal expression language such as International Phonetic Alphabet (IPA) or by transliteration into the appropriate script of the annotator's native language.

Word and sentence meaning. The better the annotator understands the full meaning of the text being annotated, the easier it will be both to identify which named entities are likely to be mentioned in the text and what the boundaries of those mentions are. This meaning can be conveyed in a variety of ways: dictionary lookup to provide fixed meanings for individual words and phrases; description of the position of a word or phrase in a semantic space (e.g., Brown clusters or embedding space) to define words that are not found in a dictionary; and full sentence translation to cue the annotator about the presence of named entities and the general meaning of each sentence.

Word context. Understanding how a word is used in a given instance can benefit greatly from understanding how

[1]https://github.com/iscoe/dragonfly


Dragonfly: IL3_NW_0200

localhost:5000/?index=2

Incognito

IL3_NW_020003.conll.txt

PER ORG GPE LOC DEL FIND

☑ Cascade

21

TOKEN	بۇ	يەردە	،	ئەنقەرە	شەرقىي	تۈركىستان	مەسىلىسىگە	خىتاينىڭ	زېمىن	نلۆكنى
ROMAN	bu	erde	,	enqere	sherqiy	tuerkistan	mesilisige	xitayning	zemin	puetue
TRANS	this			Ankara	Eastern Turki…	***			agricultural …	
BROWN	this …	way | look |…		Bangkok | Kiev | …	eastern | orie…	Gobi | Turkes…		Togo	Chatqal (village…	mediur

22

TOKEN	بۇ	نۇقتىدا	،	ئەنقەرە	ئۇيغۇر	كىملىكى	ۋە	تېخىمۇ	كەڭ	بولغان	خىتاي
ROMAN	bu	nuqtida	,	enqere	uyghur	kimliki	we	teximu	keng	bolghan	xitay
TRANS	this			Ankara	Uyghur | Uygu…		and	very | more…	broad …		China …
BROWN	this …	floor | layer | …		Bangkok | Kiev | …	Uyghur | Repu…		and	more and more…	broad …		China …

Wikipedia zemin

Notes

ning: maybe possession or from

Chai **Zemin** - Wikipedia
https://en.wikipedia.org/wiki/Chai_**Zemin**
Chai **Zemin** was a Chinese diplomat, and the first ambassador of the People's Republic of China to the United States after the normalization of the Sino-US ...

Zemin (given name) - Wikipedia
https://en.wikipedia.org/wiki/**Zemin**_(given_name)
Zemin (澤民) is a Chinese given name. People with this name include: Au Tak (also Au Chak-mun; 1840–1920), Hong Kong entrepreneur; Mao **Zemin** ...

Figure 1: A Uyghur document being annotated with Dragonfly. In this example, the Uyghur token for Ankara could be tagged based on its transliteration as "enqere", its translation from the lexicon as "Ankara" or by the Brown cluster including "Bangkok" and "Kiev". Any entry that ends in an ellipsis has more information that can be viewed by mousing over it. Tokens that are outlined have been added to the annotator's personal translation dictionary.

that word is used broadly, either across the document being annotated, or across a larger corpus of monolingual text. For example, knowing that a word frequently appears adjacent to a known person name suggests it might be a surname, even if the adjacent word in the current context is not known to be a name.

Regional knowledge. Knowledge of some of the entities, relations, and events referred to in the text allows the annotator to form a stronger model of what the text as a whole might be saying, leading to better judgments about components of the text. For example, displaying a map of city mentioned in the text could help an annotator to also find mentions of streets and buildings. Reading a Wikipedia article about a mentioned political party could lead to discovering the names of prominent politicians in the text.

Memory. It is difficult for annotators to remember previously tagged entities or insights about the grammar of the language gained from looking at the text. Programmatic support for capturing prior conclusions (linguistic patterns, word translations, possible annotations for a mention along with their frequencies) and making them readily available to the annotator is essential for large annotation efforts. Further, ordering the documents so that ones that mention the same entities are annotated consecutively helps annotators immediately apply the knowledge just gained from previous documents.

Collaboration. Different annotators notice different entities or patterns. Providing easy ways to share this information helps annotators improve quickly. Being able to adjudicate annotations on the same documents not only leads to higher quality annotations, but is an excellent way for annotators to learn from each other.

3. Design

Dragonfly takes a word-centric approach to annotation. As shown in Figure 1, each sentence to be annotated is laid out in a row with each column showing a word augmented with a variety of information about that word. Given rich enough information about the words in the sentence, it is often possible to annotate the sentence for named entities without fully comprehending the sentence.

Dragonfly renders tab-separated files (TSV) for annotation. This allows researchers to seamlessly add new semantic or syntactic features without modifying any code. In our annotation efforts, we used four lines per word. The top entry is the word in its original script. The second is a transliteration into Latin script using the ISI Universal Romanizer (Hermjakob et al., 2018). The third entry contains dictionary-based translations. The fourth is a set of dictionary translations of words found in this word's Brown cluster (Brown et al., 1992). The Brown cluster translations provides clues to the semantics or usage of the word. For example, a Brown

cluster containing translations such as "Paris", "Rome" and "Vienna" is likely to refer to a city, even if no direct translation exists to indicate a particular city.
Efficiency was an emphasis in Dragonfly's design. To reduce the number of user actions required to tag an entity mention, annotators select an entity class using the label buttons at the top of the tool (as seen in Figure 1) or through a keyboard shortcut; they can then click the words or phrases that belong to that class. Optionally, those tagging decisions can be automatically propagated throughout the document. Another efficiency-related design decision was to include search tools in the bottom panel. These are used to investigate potential entities using: a concordance search over the document collection; bilingual dictionary search; and integrated search over Wikipedia, Geonames, and Google Maps. We found that giving users direct access to these resources increases their use by annotators (increasing accuracy), and reduces the time required to interact with these services. This search capability is a new feature added to Dragonfly.

4. Annotation Procedure

Rather than enumerating all the features of Dragonfly, we describe below an annotation procedure that has worked well for us. This procedure highlights how the tool's advanced features can be used together to create high-quality annotations.

4.1. Step 1: Start with Entity-Rich Documents

Not all documents are created equal. Some are rich with well-known entities while others can be long dissertations on abstract topics. Rather than working through the document collection in order, we have found that it is more effective to start with documents that have entities that we already recognize. This is accomplished through Dragonfly's document recommendation tool, a new feature added in this version. This tool collects a list of entity names in the language of interest from sources like the Geonames gazetteer or Wikipedia. There will likely not be thousands or even hundreds of names for low resource languages, but there will likely be enough to seed the recommendation system. It is advisable to turn on the document length penalty so that most annotated documents will be in the 10 to 30 sentence range. Annotating very long documents prevent annotators from working with a wide variety of topics and entity types.
The annotators should now begin working through the recommended list of documents. The easiest entities to find are those with obvious word translations or those with Brown clusters that contain many named entities. By trying to pronounce some of the transliterations, the names of countries, cities, or famous people might be recognized. Annotation will proceed slowly at first as the annotators become familiar with the language.

4.2. Step 2: Update Translation Dictionary

Annotators will be discovering new entities and words that indicate the possible presence of entities (for example, seeing the word for *general* or *doctor* might mean a person's name is there). These words should be added to the translation dictionary built into Dragonfly by right clicking on the token and entering a translation. Tokens that are in the dictionary are outlined to draw the annotator's attention to them. Dragonfly also monitors which tokens are being tagged a high percentage of the time and highlights those on future documents. These features reduce the amount of new knowledge annotators have to remember as they progress from document to document.

4.3. Step 3: Investigate Entities Using Search

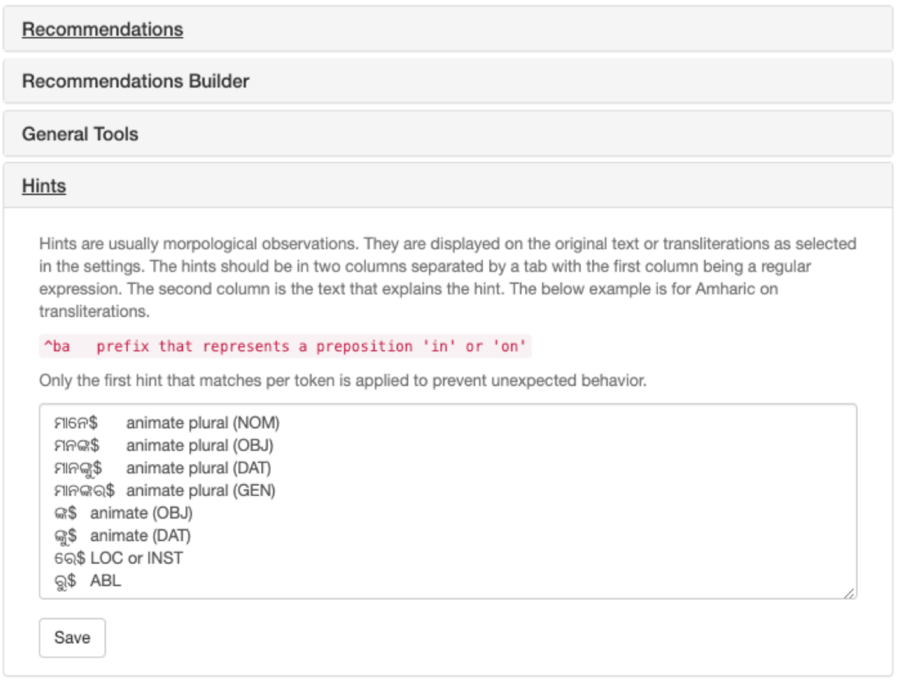
Some tokens will look like possible entities based on the surrounding context or their Brown clusters, but there may be doubt about tagging them or it may be difficult to determine their entity types. The search panel includes several tools for investigating these potential entities. Dragonfly builds a reverse index over the document collection to support a concordance search. Often seeing a word in multiple contexts helps with determining whether it is likely to be part of a name.
Dragonfly also includes search for English Wikipedia, Geonames, Google Maps, and Google Search. All of these resources have auto-suggestions or fuzzy search, which are useful when querying with transliterations. In Figure 1, the annotator searched for "zemin" (which might have appeared to be a village from its Brown cluster), but a Wikipedia search showed that it is likely a person's name. If the document collection is from a particular part of the world, Geonames and Google Maps can be constrained or biased to those locations. The combination of autosuggest, location bias, and transliterations is often successful in finding information on the entity. These resources can also be used to find the names of other entities that might be in the document.

4.4. Step 4: Add Hints

As annotators become more familiar with the language, they will begin noticing prepositions or affixes that represent directional information (the equivalent of "to", "from", "into") or other grammatical features like honorific markers. This version of Dragonfly includes a new hint capability based on regular expressions. The annotator defines the regular expression and its associated hint text. Dragonfly will then highlight the matched words with the hint text available as a tooltip.

4.5. Step 5: Share Knowledge and Reannotate

After several documents have been completed by each annotator, they should exchange their personal translation dictionaries and hints. They should then reannotate their documents, and will likely find entities that they missed. They will see new entities because of the new hints from their colleagues, but also because they have become much more proficient with the language.
Another way to share knowledge is to exchange documents and have annotators look over each others' labels. If more than one annotator worked on a document, Dragonfly also provides an adjudication mode that makes it easy to scan the annotations and find disagreements. Finally, this is also



Hints are usually morphological observations. They are displayed on the original text or transliterations as selected in the settings. The hints should be in two columns separated by a tab with the first column being a regular expression. The second column is the text that explains the hint. The below example is for Amharic on transliterations.

```
^ba   prefix that represents a preposition 'in' or 'on'
```

Only the first hint that matches per token is applied to prevent unexpected behavior.

```
ମାନେ$      animate plural (NOM)
ମନଙ୍କ$      animate plural (OBJ)
ମାନଙ୍କୁ$     animate plural (DAT)
ମାନଙ୍କର$  animate plural (GEN)
ଙ୍କ$   animate (OBJ)
ଙ୍କୁ$   animate (DAT)
ରେ$ LOC or INST
ରୁ$   ABL
```

Figure 2: Dragonfly supports regular expression-based hints. Annotators encode rules based on their observations or from looking at a grammar sketch. Words that match the regex are highlighted, with the hint text available as a tooltip on mouseover. The regexes can be defined on the native script or on the Latin transliterations. In this example, the hints are for the Odia language.

a good time to train a tagger on the annotations and add the tagger's output to the TSV files being annotated.

4.6. Repeat Process

We have found that the first few documents seem very difficult to annotators; then, there is a transition when the annotation task feels reasonable. Once the annotators have built up their knowledge of the language and region, translation dictionaries, and hints, they can build new documentation recommendations. If looking for locations, adding words for "river", "mountain", "lake", "park", "road", and "street" are a good seed set. A set of honorifics or the words for "said", "stated", and "responded" can help find documents that mention people's names. The concordance is another excellent way to find documents. After finishing a document, many of its entities and topical words are fresh in the annotator's memory. The annotator can then check the concordance for other documents that mention those topics and entities and immediately apply that knowledge to a related document.

5. Experiments

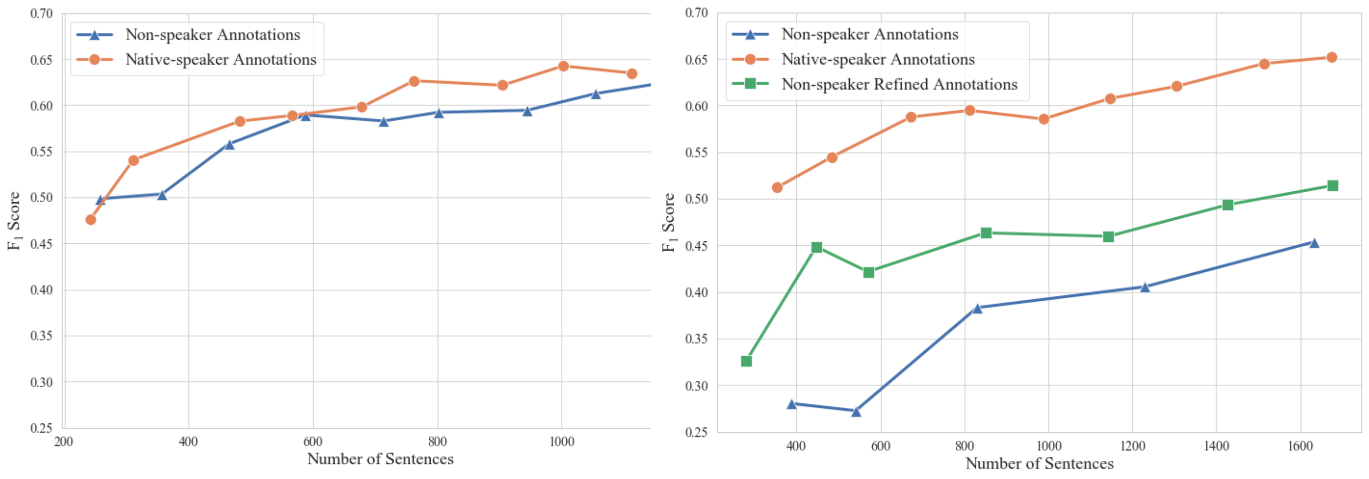
Non-speaker annotation is not perfect; it requires annotators to sometimes guess based on one or more weak clues. But is the quality of the annotations high enough to train NER models? In (Lin et al., 2018) we showed that it is possible to train NER models for multiple languages in a low resource setting using non-speaker annotation. In a five language collection of Voice of America (VOA) news, models were trained with F_1 scores that range from 55 to 76 on only approximately 500 to 2000 sentences across the languages.

In the LoReHLT 2019 Evaluation the two surprise languages were Odia (an Indian language) and Ilocano (a Filipino language). In the evaluation, we were given access to native speakers in each language for five hours and used this time to collect NER annotations. In parallel, we had three non-speakers annotate Odia documents and two non-speakers annotate Ilocano documents using Dragonfly. The setup of the annotation tool was the same as demonstrated in Figure 1. We trained a name tagger using progressively more data for each language under each annotation strategy. The tagger is written by Liu et al. (2018) and is an implementation of a bi-directional long short term memory (LSTM) network with a Conditional Random Fields (CRF) layer (Lample et al., 2016).

The Ilocano results are shown in Figure 3a. Ilocano is written in the Latin script and borrows many words from English and Spanish. As such, it was a relatively easier annotation task for non-speakers and there was only a small performance difference between native and non-speaker annotations.

Odia is more challenging to annotate than Ilocano because of its script and morphology. As can be seen in Figure 3b, there is a consistent 15 point gap in F_1 between the model trained on native speaker annotations (the orange line) and the one trained on non-speaker annotations (the blue line). Late in the evaluation, we revised a portion of the documents for Odia, which led to a five point increase in F_1 as shown by the green line in Figure 3b.

There are some caveats to these experiments on native and non-speaker annotations, which were conducted within the context of the LoReHLT evaluations. First, because of the



(a) Ilocano Language

(b) Odia Language

Figure 3: Comparison of native speaker to non-speaker annotation for NER on two languages.

limited time with the native speakers, we had them annotate entity-rich sentences. Because they understood the language of the text, we assumed that they would not need the same amount of context as the non-speakers annotating full documents. This means the training sets are not completely analogous between the two annotation approaches. Second, the native speakers were not thoroughly trained on the annotation guidelines due to the time constraints of the evaluation. This likely caused in a decrease in F_1 scores for the native speaker models. Finally, there was some exposure to the native speaker annotations by the non-speaker annotators, which likely boosted their accuracy by a small amount.

Given this experiment and our experience with previous low resource annotation efforts, we expect that the gap in performance between native speakers and non-speakers to depend on the language and the type of resources available. A language with a more extensive bilingual lexicon or a language in Latin script with a grammar similar to the non-speaker's native language will require less effort to annotate with higher quality annotations. An interesting experiment that we have not conducted is to collect time statistics on native and non-speaker annotators to measure relative speed and the extent to which non-speakers improve during the annotation process.

6. Conclusions

While at first glance non-speaker annotation seems unlikely to succeed, we and others (Mayhew and Roth, 2018; Lin et al., 2018) have demonstrated that it is possible for multiple low resource languages. Not only is non-speaker annotation possible, but the performance of NER models trained on non-speaker annotations compares well to native speaker annotations, subject to the described caveats. Use of a specialized annotation tool is critical in this setting; this paper has described the key features that any such tool must exhibit. The Dragonfly annotation tool satisfies these requirements while also being designed for efficient tagging. Finally, we have described an annotation process using Dragonfly that we have found to be effective across multiple language evaluations.

7. Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-16-C0102. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. In the event permission is required, DARPA is authorized to reproduce the copyrighted material for use as an exhibit or handout at DARPA-sponsored events and/or to post the material on the DARPA website.

8. Bibliographical References

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *COLING*.

Christianson, C., Duncan, J., and Onyshkevych, B. (2018). Overview of the darpa lorelei program. *Machine Translation*, 32(1-2):3–9.

Hermjakob, U., May, J., and Knight, K. (2018). Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia, July. Association for Computational Linguistics.

Lample, G., Ballesteros, M., Kawakami, K., Subramanian, S., and Dyer, C. (2016). Neural architectures for named entity recognition. In *NAACL HLT*.

Lin, Y., Costello, C., Zhang, B., Lu, D., Ji, H., Mayfield, J., and McNamee, P. (2018). Platforms for non-speakers annotating names in any language. In *Proceedings of ACL 2018, System Demonstrations*, pages 1–6, Melbourne, Australia, July. Association for Computational Linguistics.

Liu, L., Shang, J., Ren, X., Xu, F. F., Gui, H., Peng, J., and Han, J. (2018). Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mayhew, S. and Roth, D. (2018). Talen: Tool for annotation of low-resource entities. In *Proceedings of ACL 2018, System Demonstrations*, pages 80–86.