TALEN: Tool for Annotation of Low-resource ENtities

Stephen Mayhew
University of Pennsylvania
Philadelphia, PA
mayhew@seas.upenn.edu

Dan Roth
University of Pennsylvania
Philadelphia, PA
danroth@seas.upenn.edu

Abstract

We present a new web-based interface, TALEN, designed for named entity annotation in low-resource settings where the annotators do not speak the language. To address this non-traditional scenario, TALEN includes such features as in-place lexicon integration, TF-IDF token statistics, Internet search, and entity propagation, all implemented so as to make this difficult task efficient and frictionless. We conduct a small user study to compare against a popular annotation tool, showing that TALEN achieves higher precision and recall against ground-truth annotations, and that users strongly prefer it over the alternative.

TALEN is available at:
github.com/CogComp/talen.

1 Introduction

Named entity recognition (NER), the task of finding and classifying named entities in text, has been well-studied in English, and a select few other languages, resulting in a wealth of resources, particularly annotated training data. But for most languages, no training data exists, and annotators who speak the language can be hard or impossible to find. This low-resource scenario calls for new methods for gathering training data. Several works address this with automatic techniques (Tsai et al., 2016; Zhang et al., 2016; Mayhew et al., 2017), but often a good starting point is to elicit manual annotations from annotators who do not speak the target language.

Language annotation strategies and software have historically assumed that annotators speak the language in question. Although there has been work on *non-expert* annotators for natural language tasks (Snow et al., 2008), where the annotators lack specific skills related to the task, there has been little to no work on situations where annotators, expert or not, do not speak the language. To this end, we present a web-based interface designed for users to annotate text quickly and easily in a language they do not speak.

TALEN aids non-speaker annotators[1] with several different helps and nudges that would be unnecessary in cases of a native speaker. The main features, described in detail in Section 2, are a Named Entity (NE) specific interface, entity propagation, lexicon integration, token statistics information, and Internet search.

The tool operates in two separate modes, each with all the helps described above. The first mode displays atomic documents in a manner analogous to nearly all prior annotation software. The second mode operates on the sentence level, patterned on bootstrapping with a human in the loop, and designed for efficient discovery and annotation.

In addition to being useful for non-speaker annotations, TALEN can be used as a lightweight inspection and annotation tool for within-language named entity annotation. TALEN is agnostic to labelset, which means that it can also be used for a wide variety of sequence tagging tasks.

2 Main Features

In this section, we describe the main features individually in detail.

Named Entity-Specific Interface

The interface is designed specifically for Named Entity (NE) annotation, where entities are rela-

[1] We use 'non-speaker' to denote a person who does not speak or understand the language, in contrast with 'non-native speaker', which implies at least a shallow understanding of the language.

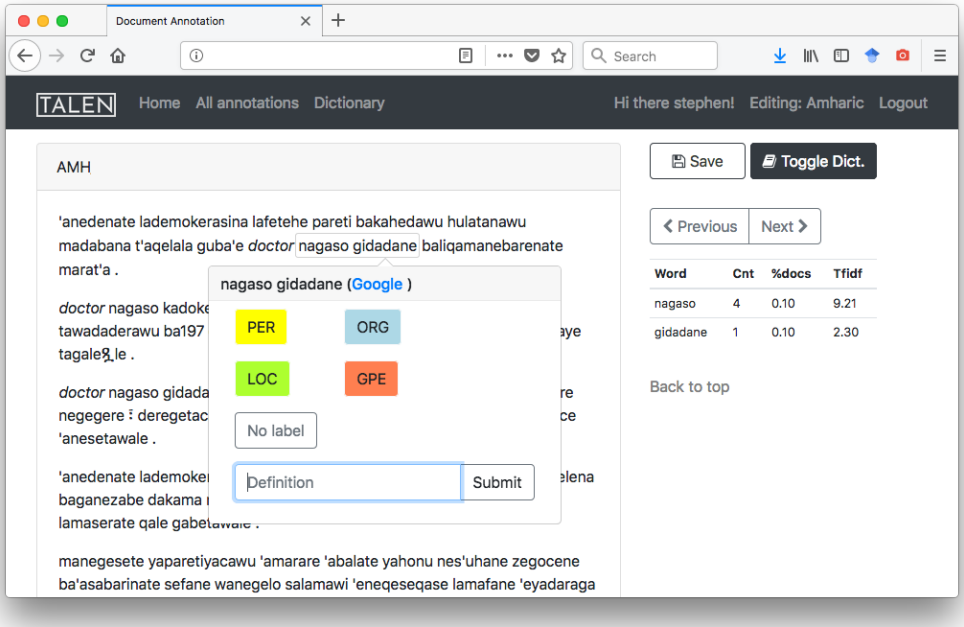

Figure 1: Document-based annotation screen. A romanized document from an Amharic corpus is shown. The user has selected "nagaso gidadane" (Negasso Gidada) for tagging, indicated by the thin gray border, and by the popover component. The lexicon (dictionary) is active, and is displaying the definition (in italics) for "doctor" immediately prior to "nagaso". (URL and document title are obscured).

tively rare in the document. The text is intentionally displayed organically, in a way that is familiar and compact, so that the annotator can see as much as possible on any given screen. This makes it easy for an annotator to make document-level decisions, for example, if an unusual-looking phrase appears several times. To add an annotation, as shown in Figure 1, the annotator clicks (and drags) on a word (or phrase), and a popover appears with buttons corresponding to label choices as defined in the configuration file. Most words are not names, so a default label of non-name (O) is assigned to all untouched tokens, keeping the number of clicks to a minimum. In contrast, a part-of-speech (POS) annotation system, SAWT (Samih et al., 2016), for example, is designed so that every token requires a decision and a click.

Entity Propagation

In a low-resource scenario, it can be difficult to discover and notice names because all words look unfamiliar. To ease this burden, and to save on clicks, the interface propagates all annotation decisions to every matching surface in the document. For example, if *'iteyop'eya* (Ethiopia) shows up many times in a document, then a single click will annotate all of them.

In the future, we plan to make this entity propagation smarter by allowing propagation to near surface forms (in cases where a suffix or prefix differs from the target phrase) or to cancel propagation on incorrect phrases, such as stopwords.

Lexicon Integration

The main difficulty for non-speakers is that they do not understand the text they are annotating. An important feature of TALEN is in-place lexicon integration, which replaces words in the annotation screen with their translation from the lexicon. For example, in Figure 1, the English word *doctor* is displayed in line (translations are marked by italics). As before, this is built on the notion that annotation is easiest when text looks organic, with translations seamlessly integrated. Users can click on a token and add a definition, which is immediately saved to disk. The next time the annotator encounters this word, the definition will be displayed. If available, a bilingual lexicon (perhaps from PanLex (Kamholz et al., 2014)), can kickstart the process. A side effect of the definition addition feature is a new or updated lexicon, which can be shared with other users, or used for other tasks.

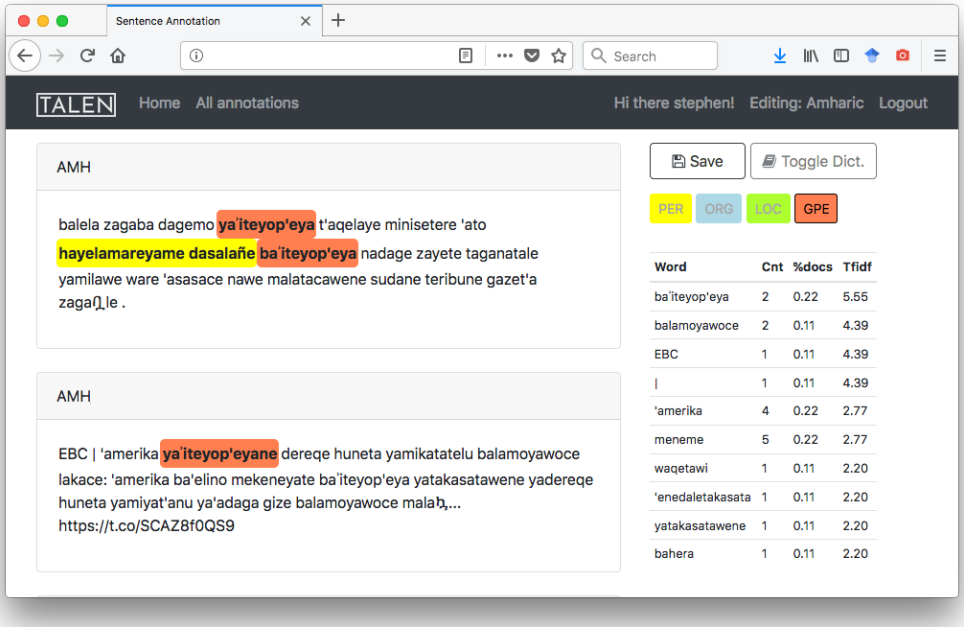

Figure 2: Sentence-based annotation screen, with sentences corresponding to seed term *ba'iteyop'eya* (annotated in the first sentence, not in the second). Two sentences are shown, and the remaining three sentences associated with this seed term are lower on the page. Notice that *hayelamareyame dasalañe* (Hailemariam Desalegn) has also been annotated in the first sentence. This will become a new seed term for future iterations. (URL and sentence IDs are obscured).

Token Statistics

When one has no knowledge of a language, it may be useful to know various statistics about tokens, such as document count, corpus count, or TF-IDF. For example, if a token appears many times in every document, it is likely not a name. Our annotation screen shows a table with statistics over tokens, including document count, percentage of documents containing it, and TF-IDF. At first, it shows the top 10 tokens by TF-IDF, but the user can click on any token to get individual statistics. For example, in Figure 1, *nagaso* appears 4 times in this document, appears in 10% of the total documents, and has a TF-IDF score of 9.21. In practice, we have found that this helps to give an idea of the topic of the document, and often names will have high TF-IDF in a document.

Internet Search

Upon selection of any token or phrase, the popover includes an external link to search the Internet for that phrase. This can be helpful in deciding if a phrase is a name or not, as search results may return images, or even autocorrect the phrase to an English standard spelling.

3 Annotation Modes

There are two annotation modes: a document-based mode, and a sentence-based mode. Each has all the helps described above, but they display documents and sentences to users in different ways.

3.1 Document-based Annotation

The document-based annotation is identical to the common paradigm of document annotation: the administrator provides a group of documents, and creates a configuration file for that set. The annotator views one document at a time, and moves on to the next when they are satisfied with the annotations. Annotation proceeds in a linear fashion, although annotators may revisit old documents to fix earlier mistakes. Figure 1 shows an example of usage in the document-based annotation mode.

3.2 Sentence-based Annotation

The sentence-based annotation mode is modeled after bootstrapping methods. In this mode, the configuration file is given a path to a corpus of documents (usually a very large corpus) and a small number (less than 10) seed entities, which

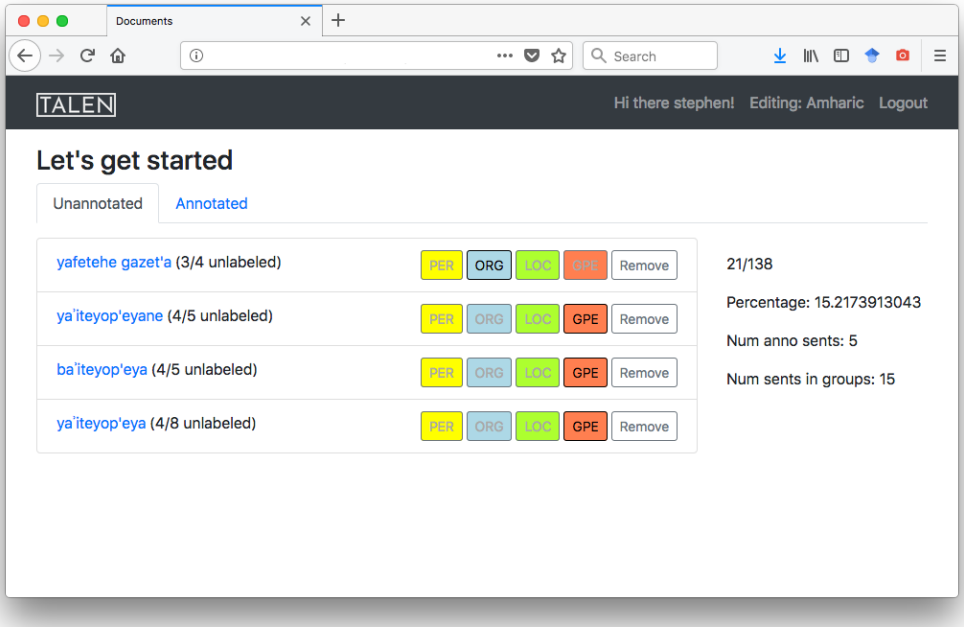

Figure 3: Sentence-based annotation screen showing 4 seed terms available for annotation. Notice the *Unannotated* and *Annotated* tabs. These terms are in the active *Unannotated* tab because each term has some sentences that have not yet been labeled with that seed term. For example, of the 5 sentences found for *ba'iteyop'eya*, only 1 has this seed term labeled (see Figure 2).

can be easily acquired from Wikipedia. This corpus is indexed at the sentence level, and for each seed entity, k sentences are retrieved. These are presented to the annotator, as in Figure 2, who will mark all names in the sentences, starting with the entity used to gather the sentence, and hopefully discover other names in the process. As names are discovered, they are added to the list of seed entities, as shown in Figure 3. New sentences are then retrieved, and the process continues until a predefined number of sentences has been retrieved. At this point, the data set is frozen, no new sentences are added, and the annotator is expected to thoroughly examine each sentence to discover all named entities.

In practice, we found that the size of k, which is the number of sentences retrieved per seed term, affects the overall corpus diversity. If k is large relative to the desired number of sentences, then the annotation is fast (because entity propagation can annotate all sentences with one click), but the method produces a smaller number of unique entities. However, if k is small, annotation may be slower, but return more diverse entities. In practice, we use a default value of $k = 5$.

4 Related Work

There are several tools designed for similar purposes, although to our knowledge none are designed specifically for non-speaker annotations. Many of the following tools are powerful and flexible, but would require significant refactoring to accommodate non-speakers.

The most prominent and popular is brat: rapid annotation tool (brat) (Stenetorp et al., 2012), a web-based general purpose annotation tool capable of a handling a wide variety of annotation tasks, including span annotations, and relations between spans. brat is open source, reliable, and available to download and run.

There are a number of tools with similar functionality. Sapient (Liakata et al., 2009) is a web-based system for annotating sentences in scientific documents. WebAnno (de Castilho et al., 2016) uses the frontend visualization from brat with a new backend designed to make large-scale project management easier. EasyTree (Little and Tratz, 2016) is a simple web-based tool for annotating dependency trees. Callisto[2] is a desktop tool for rich linguistic annotation.

[2]http://mitre.github.io/callisto/

SAWT (Samih et al., 2016) is a sequence annotation tool with a focus on being simple and lightweight, which is also a focus of ours. One key difference is that this expects that annotators will want to provide a tag for every word. This is inefficient for NER, where many tokens should take the default non-name label.

5 Experiment: Compare to brat

The brat rapid annotation tool (brat) (Stenetorp et al., 2012) is a popular and well-featured annotation tool, which makes for a natural comparison to TALEN. In this experiment, we compare tools qualitatively and quantitatively by hiring a group of annotators. We can compare performance between TALEN and brat by measuring the results after having annotators use both tools.

We chose to annotate Amharic, a language from Ethiopia. We have gold training and test data for this language from the Linguistic Data Consortium (LDC2016E87). The corpus is composed of several different genres, including newswire, discussion forums, web blogs, and social network (Twitter). In the interest of controlling for the domain, we chose only 125 newswire documents (NW) from the gold data, and removed all annotations before distribution. Since Amharic is written in Ge'ez script, we romanized it, so it can be read by English speakers. We partitioned the newswire documents into 12 (roughly even) groups of documents, and assigned each annotator 2 groups: one to be annotated in brat, the other with TALEN. This way, every annotator will use both interfaces, and every document will be annotated by both interfaces. We chose one fully annotated gold document and copied it into each group, so that the annotators have an annotation example.

We employed 12 annotators chosen from our NLP research group. Before the annotation period, all participants were given a survey about tool usage and language fluency. No users had familiarity with TALEN, and only one had any familiarity with brat. Of the annotators, none spoke Amharic or any related language, although one annotator had some familiarity with Hebrew, which shares a common ancestry with Amharic, and one annotator was from West Africa.[3]

Immediately prior to the annotation period, we gave a 15 minute presentation with instructions on tool usage, annotation guidelines, and annotation strategies. The tags used were Person, Organization, Location, and Geo-Political Entity. As for strategy, we instructed them to move quickly, annotating names only if they are confident (e.g. if they know the English version of that name), and to prioritize diversity of discovered surface forms over exhaustiveness of annotation. When using TALEN, we encouraged them to make heavy use of the lexicon. We provided a short list (less than 20 names) of English names that are likely to be found in documents from Ethiopia: local politicians, cities in the region, etc.

The annotation period lasted 1 hour, and consisted of two half hour sessions. For the first session, we randomly assigned half the annotators to use brat, and the other half to use TALEN. When this 30 minute period was over, all annotators switched tools. Those who had used brat use ours, and vice versa. We did this because users are likely to get better at annotating over time, so the second tool presented should give better results. Our switching procedure mitigates this effect.

At the end of the second session, each document group had been annotated twice: once by some annotator using brat, and once by some annotator using TALEN. These annotations were separate, so each tool started with a fresh copy of the data.

We report results in two ways: first, annotation quality as measured against a gold standard, and second, annotator feedback. Figure 4 shows basic statistics on the datasets. Since the documents

DATASET	PRECISION	RECALL	F1
brat	51.4	8.7	14.2
TALEN	53.6	12.6	20.0

DATASET	TOTAL NAMES	UNIQUE NAMES
Gold	2260	1154
brat	405	189
TALEN	457	174

Figure 4: Performance results. The precision, recall, and F1 are measured against the gold standard Amharic training data. When counting *Unique Names*, each unique surface form is counted once. We emphasize that these results are calculated over a *very* small amount of data annotated over a half-hour period by annotators with no experience with TALEN or Amharic. These only show a quick and dirty comparison to brat, and are not intended to demonstrate high-quality performance.

[3] Ethiopia is in East Africa.

we gave to the annotators came from a gold annotated set, we calculated precision, recall, and F1 with respect to the gold labels. First, we see that TALEN gives a 5.8 point F1 improvement over brat. This comes mostly from the recall, which improves by 3.9 points. This may be due to the automatic propagation, or it may be that having a lexicon helped users discover more names by proximity to known translations like *president*. In a less time-constrained environment, users of brat might be more likely select and annotate all surfaces of a name, but the reality is that all annotation projects are time-constrained, and any help is valuable.

The bottom part of the table shows the annotation statistics from TALEN compared with brat. TALEN yielded more name spans than brat, but fewer unique names, meaning that many of the names from TALEN are copies. This is also likely a product of the name propagation feature.

We gathered qualitative results from a feedback form filled out by each annotator after the evaluation. All but one of the annotators preferred TALEN for this task. In another question, they were asked to select an option for 3 qualities of each tool: efficiency, ease of use, and presentation. Each quality could take the options *Bad, Neutral,* or *Good*. On each of these qualities, brat had a majority of *Neutral*, and TALEN had a majority of *Good*. For TALEN, *Efficiency* was the highest rated quality, with 10 respondents choosing *Good*.

We also presented respondents with the 4 major features of TALEN (TF-IDF box, lexicon, entity propagation, Internet search), and asked them to rate them as *Useful* or *Not useful* in their experience. Only 4 people found the TF-IDF box useful; 10 people found the lexicon useful; all 12 people found the entity propagation useful; 7 people found the Internet search useful. These results are also reflected in the free text feedback. Most respondents were favorable towards the lexicon, and some respondents wrote that the TF-IDF box would be useful with more exposure, or with better integration (e.g. highlight on hover).

6 Technical Details

The interface is web-based, with a Java backend server. The frontend is built using Twitter bootstrap framework,[4] and a custom javascript library called `annotate.js`. The backend is built with the Spring Framework, written in Java, using the TextAnnotation data structure from the CogComp NLP library (Khashabi et al., 2018) to represent and process text.

In cases where it is not practical to run a backend server (for example, Amazon Mechanical Turk[5]), we include a version written entirely in javascript, called `annotate-local.js`.

We allow a number of standard input file formats, and attempt to automatically discover the format. The formats we allow are: column, in which each *(word, tag)* pair is on a single line, serialized TextAnnotation format (from CogComp NLP (Khashabi et al., 2018)) in both Java serialization and JSON, and CoNLL column format, in which the tag and word are in columns 0 and 5, respectively.

When a user logs in and chooses a dataset to annotate, a new folder is created automatically with the username in the path. When that user logs in again, the user folder is reloaded on top of the original data folder. This means that multiple users can annotate the same corpus, and their annotations are saved in separate folders. This is in contrast to brat, where each dataset is modified in place.

7 Conclusions and Future Work

We have presented TALEN, a powerful interface for annotation of named entities in low-resource languages. We have explained the usage with screenshots and descriptions, and outlined a short user study showing that TALEN outperforms brat in a low-resource task in both qualitative and quantitative measures.

Acknowledgements

We would like to thank our annotators for their time and help in the small annotation project.

This material is based on research sponsored by the US Defense Advanced Research Projects Agency (DARPA) under agreement numbers FA8750-13-2-0008 and HR0011-15-2-0025. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

[4] https://getbootstrap.com/

[5] mturk.com

References

Richard Eckart de Castilho, Eva Mujdricza-Maydt, Muhie Seid Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*. pages 76–84.

David Kamholz, Jonathan Pool, and Susan M Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *LREC*. pages 3145–3150.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai, Subhro Roy, Stephen Mayhew, Zhilli Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, and Dan Roth. 2018. Cogcompnlp: Your swiss army knife for nlp. In *11th Language Resources and Evaluation Conference*.

Maria Liakata, Claire Q, and Larisa N. Soldatova. 2009. Semantic annotation of papers: Interface & enrichment tool (sapient). In *BioNLP@HLT-NAACL*.

Alexa Little and Stephen Tratz. 2016. Easytree: A graphical tool for dependency tree annotation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *EMNLP*. http://cogcomp.org/papers/MayhewTsRo17.pdf.

Younes Samih, Wolfgang Maier, Laura Kallmeyer, and Heinrich Heine. 2016. Sawt: Sequence annotation web tool.

Rion Snow, Brendan T. O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*. Association for Computational Linguistics, Avignon, France.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*. http://cogcomp.org/papers/TsaiMaRo16.pdf.

Boliang Zhang, Xiaoman Pan, Tianlu Wang, Ashish Vaswani, Heng Ji, Kevin Knight, and Daniel Marcu. 2016. Name tagging for low-resource incident languages based on expectation-driven learning. In *NAACL*.