Employing Wikipedia as a Resource for Named Entity Recognition in Morphologically Complex Under-Resourced Languages

Aravind Krishnan[1,3], Stefan Ziehe[2], Franziska Pannach[3], and Caroline Sporleder[2,3]

aravindhp1999@gmail.com,
{caroline.sporleder,stefan.ziehe}@cs.uni-goettingen.de,
franziska.pannach@uni-goettingen.de

[1]College of Engineering Trivandrum, India
[2]Institute of Computer Science, University of Göttingen, Germany
[3]Göttingen Centre for Digital Humanities, Germany

Abstract

We propose a novel approach for rapid prototyping of named entity recognisers through the development of semi-automatically annotated data sets. We demonstrate the proposed pipeline on two under-resourced agglutinating languages: the Dravidian language *Malayalam* and the Bantu language *isiZulu*. Our approach is weakly supervised and bootstraps training data from Wikipedia and Google Knowledge Graph. Moreover, our approach is relatively language independent and can consequently be ported quickly (and hence cost-effectively) from one language to another, requiring only minor language-specific tailoring.

1 Introduction

Named entity recognition (NER) is the task of identifying proper names and assigning them to one of several named entity (NE) classes, such as PERSON (PER), LOCATION (LOC) or ORGANISATION (ORG), which is a crucial processing step for many NLP tasks, but also for many applications in the digital humanities where information about the entities involved (e.g. names of emperors or archaeological sites) is often particularly important. While state-of-the-art systems obtain good results for standard NE inventories and general purpose English (Chiu and Nichols, 2016), annotated data sets for the development of named entity taggers are not readily available for most of the world's languages.[1]

In this paper, we focus on semi-automatically generating annotated data and bootstrapping NE recognisers for under-resourced languages (cf. Krauwer (2003)), i.e., languages for which manually annotated data as well as pre-processing tools, such as part-of-speech taggers, are typically hard to come by. To this end, we propose a *weakly supervised* approach that bootstraps the training set from Wikipedia (in the target language) and Google knowledge graph (in English), requiring no manual annotation and no pre-processing apart from the language-specific tweaking of our matching heuristics. This approach is therefore in principle suitable for any language for which Wikipedia articles exist.[2] Because the manual effort is limited, systems can be quickly ported to new languages, while still obtaining reasonable results.

We demonstrate this by developing the system for *Malayalam* and then porting it to *isiZulu*. These two languages were chosen because they are agglutinating and morphologically complex, making the task considerably more challenging than for many Indo-European languages where NEs are only minimally inflected. While our target languages are both agglutinating, they are also structurally quite different in other respects and exhibit different degrees of "under-resourcing", with noticeably fewer resources being available for isiZulu (see Sect. 3)

2 Related Work

Wikipedia has been employed for NER in three main ways: In a monolingual setting, early studies used it to extract Gazetteer lists which were then used as features in (typically supervised) NER systems. One of the first studies taking this approach was by Toral and Muñoz (2006), who extract Gazetteers by matching the first sentence of a Wikipedia article heuristically against the WordNet

[1]Even for English, NER is not necessarily a solved problem for specialised domains, which often require specific entity class inventories (Brandsen et al., 2020).

[2]As of July 2021 this applies to 323 languages. Arguably this still leaves out a large amount of the world's 6000+ languages but it covers many languages which have a fair amount of speakers but are still under-resourced. Furthermore, Wikipedia is constantly growing both in terms of content for a given language and in terms of the languages it covers.

Proceedings of 14th Workshop on Building and Using Comparable Corpora in conjunction with RANLP-2021, pages 28–39, held online, 6 September 2021.
https://doi.org/10.26615/978-954-452-076-2_005

(Fellbaum, 1998) noun hierarchy to identify the category of the entity described. This was followed by a number of similar approaches (Kazama and Torisawa, 2007; Ratinov and Roth, 2009; Radford et al., 2015).

Going one step further, some researchers used Wikipedia not only for extracting Gazetteers but also for bootstrapping annotated training data. For example, Nothman et al. (2008) exploit hyperlinks to annotate the sentences containing them with category information, which is extracted from the article the hyperlink links *to*. As not all mentions of an entity in an article are hyperlinked, they extend the data set by finding verbatim repetitions of the hyperlink's anchor text in the article. Finally, they use the data to train an NE tagger. The system requires hand-labelling of seed data that maps information extracted from articles to NE classes.

Wikipedia has also been used in a multilingual setting to obtain NE taggers for languages other than English, e.g. by exploiting cross-lingual links between articles (Richman and Schone, 2008; Bhagavatula et al., 2012; Pan et al., 2017). This approach has also been applied to under-resourced languages (Littell et al., 2016). Ni and Florian (2016) go one step further and construct entity type mappings for the English Wikipedia before projecting across Wikipedia language links.

Bouamor et al. (Bouamor et al., 2013) propose employing Wikipedia as a resource for creating domain-specific lexicons for machine-translation. They demonstrate their approach for English-French and English-Romanian translation tasks. Mayhew et al. (Mayhew et al., 2017) combine lexicon-based translation of training data from a source to a target language with features generated from Wikipedia and show that this approach can be applied to under-resourced languages.

Studies that address NER for our target languages are very limited. To our knowledge the first NER system for Malayalam was proposed by Bindu and Idicula (2011), who use supervised machine learning utilising a variety of features complemented with a finite-state automaton to deal with complex words. Jayan et al. (2013) propose a hybrid approach that combines rules with supervised machine learning. Devi et al. (2016) tackle named entity extraction from social media and combine supervised machine learning (SVMs) with skip-gram features. Shruthi and Pranav (2016) propose another supervised approach based on the TnT tagger (Brants, 2002) and maximum entropy models. A neural network approach is proposed by Ajees and Idicula (2018) who use word embeddings of context words and morphs of the target word as features. A similar system but with a different neural architecture (RNN-LSTM) has also been proposed (Sreeja and Pillai, 2020). To our knowledge, the only NER system for isiZulu was proposed by Eiselen (2016), who used linear-chain Conditional Random Fields (CRFs) for the classification of the named entities. The features included gazetteer lists and graphemic information (capitalization, punctuation, numerals).

3 The Target Languages: Malayalam and isiZulu

We test our system on two agglutinating languages: Malayalam and isiZulu. We hypothesise that inflection and agglutination will make the task particularly challenging, as one token can correspond to several linguistic words (see Sec. 3.1 and 3.2). However, Malayalam and isiZulu also differ in several aspects: They use different writing systems (Brahmic vs. Latin) and while the former tends to make extensive use of suffixes the latter tends to favour prefixes to encode grammatical information. From a practical perspective, while both languages are under-resourced, isiZulu is so to a greater extent, in particular its Wikipedia version is more than an order of magnitude smaller (see Sec. 4). We thus believe that these two languages pose sufficiently heterogeneous use cases.

3.1 Malayalam

Malayalam is the official language of the Indian state of Kerala. It is a Dravidian language and shares its roots with other south Indian languages such as Tamil and Telegu. Malayalam is spoken by 45 million people, mainly in Kerala, Lakshadeep and Puducherry. Like most Dravidian languages, Malayalam has a Subject-Object-Verb canonical order. It is a heavily agglutinating language. Finite verbs in Malayalam are inflected based on tense and mood, and are invariant to gender or number. Inflection is usually carried out through suffixing. A noun in Malayalam can be suffixed in at least 7 different ways according to the case and grammatical category employed.

For example, "Kochi" (കൊച്ചി) is a place in Kerala. കൊച്ചിയിൽ means "*inside/in Kochi*". കൊച്ചിയിൽനിന്നും means *from Kochi* and കൊച്ചിയുടെ means *of Kochi*. The word can be be inflected in various other ways as well. An example of suffixing within a sentence is depicted in (1).

(1) ഹനുമാൻ സീത + യെ കാണുവാൻ
Hanuman Seetha + accusative to see
ലങ്ക + യിലേക്ക് പോയി
Lanka + to go
'Hanuman went to Lanka to see Seetha'

Agglutination is optional in Malayalam. Therefore, a word has the option of merging with another consecutive word, producing a new word in the process. For example, കൊച്ചിയിൽ ആയിരുന്നു (കൊച്ചിയിൽ: in Kochi, ആയിരുന്നു: was) translates to *was in Kochi*. The two words can be optionally combined into a new token: കൊച്ചിയിലായിരുന്നു (*was in Kochi*). Grammatically speaking, the split version and the agglutinated version can be used interchangeably in a sentence. This increases the complexity of token matching and dictionary generation significantly. Furthermore, unlike languages written in Latin script, Malayalam does not distinguish between upper and lower case in its writing system, hence casing cannot be used as a cue for named entity recognition.

Although it is an under-resourced language, the presence of Malayalam in the form of articles and data repositories on the internet has been growing steadily over the years. It has featured in a limited number of NLP tasks, including morphological analysis (Bhavukam et al., 2018), POS tagging (Akhil et al., 2020) and NER (Ajees and Idicula, 2018). However, many studies use small locally generated data sets (Nambiar et al., 2019) or domain specific data sets (Kumar et al., 2019), (Devi et al., 2016), which usually are not freely available.

3.2 IsiZulu

IsiZulu is the language of the Zulu people in Southern Africa. It is spoken by approximately 10.6 Million people (Taljard and Bosch, 2006), mainly in the eastern part of South Africa and Mozambique. IsiZulu is an agglutinating, conjunctively-written language and belongs to the Bantu languages (Nguni sub-branch) (Taljard and Bosch, 2006). As is characteristic for Bantu languages, isiZulu uses noun classes, e.g. dedicated classes for nouns describing humans in singular or plural.

Certain natural language processing tasks can be very challenging or almost infeasible to solve for languages such as isiZulu. For instance, due to the nature of isiZulu concords, prefixes and infixes[3], sentences might consist of ambiguous words, as in Example (2). Another characteristic that isiZulu shares with other conjunctive languages is the use of capitalization inside a word, which can be an indicator of a named entity, e.g. *eGoli* – in/from Johannesburg, as in (3). Cultural naming conventions are another challenge for NER (Eiselen, 2016). For example, *Nkosi* means *king, lord* or *chief* and can be both first- or lastname, as for the South African rugby players S'busiso Nkosi and Nkosi Nofuma.

(2) Aba+ fundi a+
CLASS-2-NOUN-PREFIX-PL. learn NEG
ba+ fund+ i.
SUBJ-CONDORD learn NEG
'The students are not learning.'

(3) Umfo+ wethu u+
brother 1ST-PERS-POSS SUBJ.-CONCORD
hlala e+ Goli.
stay LOC Johannesburg
'My brother stays in Johannesburg.'

While isiZulu is not an endangered language[4], there is a lack of large digital textual resources, such as newspaper archives, and consequently also of NLP tools. The South African Centre for Digital Language Resources (SADiLaR) is one of the main drivers of language development in South Africa. Besides their teaching and knowledge sharing efforts, SADiLaR also collects resources for the South African languages and makes them available through their website[5]. The SADiLaR repository currenty lists 49 language resources, tools and corpora for isiZulu.

4 Data Sets and Resources

For bootstrapping **training** data, we utilise the **Malayalam** and **isiZulu Wikipedias**. The former is significantly larger (65,000 vs. 2,701 articles as of June 2020) and therefore also gives rise to a larger training set. In order to find appropriate named entity tags for Wikipedia articles (see Sect. 5.2) we employ the **Google Knowledge Graph**

[3] https://en.wiktionary.org/wiki/Category:Zulu_prefixes
[4] https://glottolog.org/resource/languoid/id/zulu1248
[5] https://www.sadilar.org/index.php/en/

(GKG) (Singhal, 2012). We **test** our system on two external data sets for Malayalam (ARNEKT and CUSAT) and one for isiZulu (NCHLT II) as well as part of our bootstrapped data:

ARNEKT IECSIL FIRE 2018 NER Dataset This corpus was compiled from the abstracts and info-box properties from DBpedia for the (IECSIL) shared task (Hullathy Balakrishnan et al., 2018). The info-box features are used to annotate long abstracts. Meta tags are translated into English using Google translator. The data set consists of 838,333 tokens overall: 59,422 PER, 29,371 LOC, and 4,841 ORG. All other tokens are labelled OTHER.

CUSAT NER Dataset This is a manually annotated NER data set developed by CUSAT.[6] It is based on the CUSAT POS tagged data set for Malayalam (Ajees and Idicula, 2018). About 200,000 words from "internet texts" were manually annotated. The POS tags were ignored and the data was cleaned to remove special characters. The data set consists of 190,265 tokens overall, with 1,864 PER, 1,035 LOC, and 496 ORG entities. It is thus considerably smaller than the ARNEKT data set.

NCHLT II Dataset This isiZulu data set consists of South African governmental texts, which are manually annotated with named entities (Eiselen, 2016), containing 5,024 PER, 3,872 LOC, and 5,039 ORG, 1,8224 MISC (i.e. other entity classes), and 169,393 OUT (non-entities) tokens. For evaluation, we merge the latter two classes to OTHER.

WikiML and WikiZu Apart from ARNEKT, the above data sets come from other domains as our training data (Wikipedia). Hence, testing on them can be seen as an out-of-domain lower bound evaluation of our system. For comparison, we therefore also test on a 10% portion of our Wikipedia data sets (see Sect. 5). This constitutes an upper bound as these data sets are from the same domain as the training data but are labelled automatically in a fashion identical to labelling the training data, which might lead to overly optimistic results.

5 Bootstrapping the Training Data

As our focus is on under-resourced languages, we do not assume that a manually labelled training set is available. Instead we bootstrap from Wikipedia and GKG. Utilising Wikipedia has a number of advantages: First, as it is community-driven, many under-resourced languages have a version of Wikipedia. Second, Wikipedia articles cover a wide range of subjects and often refer to named entities. Third, Wikipedia has a number of features that help with bootstrapping entity labels (see Sect. 2). Finally, it has been shown that additional training data bootstrapped from Wikipedia can also improve the performance of taggers trained on other sources, especially if they are applied out-of-domain (Nothman et al., 2009).

We employ a 4-step pipeline to bootstrap NE labelled data (Fig. 1): First, we extract a list of titles from Wikipedia dumps in the target language. Second, we use the Wikipedia language links to look up their English counterparts. Third, we employ the GKG to extract candidates for named entity tags. Finally, we use the title list to annotate Wikipedia articles. The distribution of the different NE tags for both data sets is shown in Table 1.

NE Tag	Malayalam	isiZulu
Other	21012137	138986
Place	723259	5916
Person	444260	2748
Organization	179022	700
Total	**22358678**	**148350**

Table 1: NE token distribution, compiled data sets

5.1 Creation of the Title Lists

We compile a list of all article titles from the Wikipedia dump of the target language[7] and preprocess it by removing all entries that do not contain at least one character in the target language. This removes titles entirely composed of numbers, special characters and characters from other languages. Duplicate titles are also removed. The title list includes titles which share the primary token, but contain descriptors in brackets to distinguish them, for example ഉണ്ണിയാർച്ച: *Unniyarcha* and ഉണ്ണിയാർച്ച (ചലച്ചിത്രം): *Unniyarcha (Film)*, where the descriptor helps distinguish the person from the movie. Descriptors are preserved, because they are vital when annotating the title with an NE tag.

[6] https://www.cusat.ac.in/

[7] For Malayalam, we used a Wikipedia dump from July 2020, for isiZulu from January 2021.

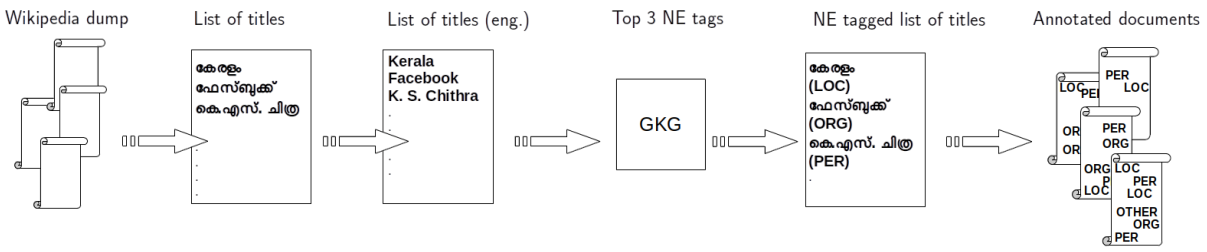


Figure 1: Schematic overview of the data set generation process

5.2 Labeling Titles with NE Tags

In order to assign titles their respective NE tags, we query each title in the title list through the GKG, which associates the search result with a tag similar to entity tags used in NER systems. For the purpose of this study, the tags have been limited to PER, LOC and ORG, since they are the most widely employed entity types. Entities that do not fall into these categories are labelled OTHER. As the GKG accepts only English queries, we need to translate (and transliterate for Malayalam) titles from the target languages. We exploit the multilingualism in Wikipedia to map titles from the target language to their respective counterparts in English.

The GKG makes use of different sources when producing tags and will generate a ranked list of (possibly different) tags for each query. We consider only the top three of these. If one of the three named entity tags appears in this list, it is assigned to the respective title, with priority being given to the higher ranking source. If the tags generated by the GKG do not contain any of our named entities, the title is annotated with the tag OTHER (i.e. no named entity or a named entity belonging to a different category such as DATE). We then automatically annotate the text of each Wikipedia article, assigning each token one of three NE tags (PERS, LOC, ORG) or the tag OTHER. As illustrated in Figure 2, we perform two "sweeps":

The first stage of the first sweep exploits hyperlinks to annotate tokens within an article. Even if a title present in the body of the article is ambiguous, a hyperlink will direct to the correct source and tag. For example, tokens that have different NE tags but the same primary token, e.g. *Unniyarcha* and *Unniyarcha (Film)*, can be disambiguated by extracting the corresponding named entity tag for each hyperlink from the title list created earlier. Then, the descriptions within brackets are removed in the case of ambiguous titles. All appearances of hyperlinks are annotated with their respective tags. Tokens that do not match any hyperlink are labelled OTHER.

In the second stage of the first sweep, all occurrences of titles that are not hyperlinked in the article body are annotated. For each article, the tokens labelled OTHER after the first stage are compared with the named entity titles in the title list. All token matches are annotated with the tag of the respective title.

In the second sweep, we annotate tokens that match sub-words of named entity titles in our title list, i.e., we annotate inflected forms and complex words. This is necessary because, in agglutinating languages, proper nouns seldom exist in their base form. This makes the matching of words that refer to the same concept harder than for languages such as English, which only has minimal pre- and suffixing, because a simple search for string equality with a title will not suffice. Therefore, we developed language-specific token-title matching algorithms discussed in the next sections. Since the secondary sweep is executed only after the ambiguous tokens are dealt with, the annotation procedure tackles both reliability and quantity of annotations.

5.3 Accommodating Morphological Characteristics

5.3.1 Heuristics for Malayalam

Suffix matching A major problem for NER in Malayalam is that —due to inflection and agglutination— nouns rarely occur in their base form but are typically adorned by suffixes. To solve this problem, a suffix stripping algorithm is employed, which initially compares each title in the list with the tokens in the body of the article and extracts all tokens that qualify a basic distance match. A threshold of 70% match was empirically found to work well. To counteract overgeneration and ensure the presence of suffixation, the results are

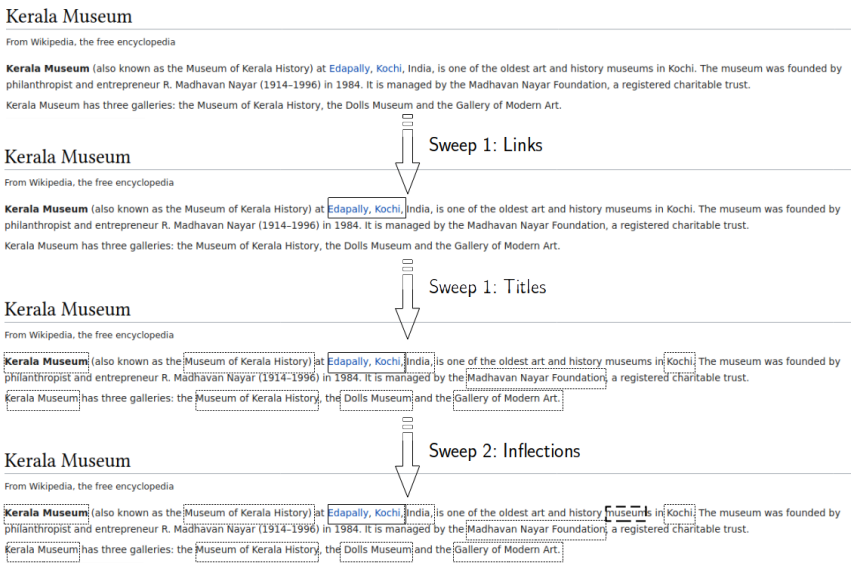


Figure 2: Overview of the three stages of data set annotation

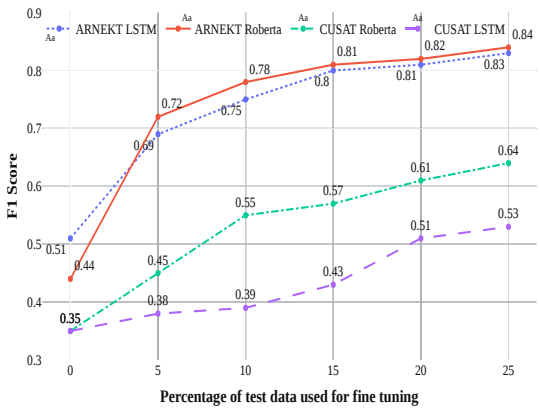
further filtered by checking if the token begins with the root word. That is, the first $(n-1)$ characters of the token must match the first $(n-1)$ characters of the title, for a title of length n. This separates suffixed versions from accidental matches. For example, the title പന്തളം (Panthalam-*Place*) matches both പന്തയം (Panthayam-*competition*) and പന്തളവും (Panthala+vum-*Panthalam as well*). Only the second token is an inflected version. The suffix match with the first $(n-1)$ characters ('പ', 'ന്ത', 'ള') extracts the inflected token and discards arbitrary matches.

Attachment of the place of origin to a person's name It is a common practice in Kerala to attach the place of a person's origin to their name. For example, consider the name Pinarayi Vijayan (പിണറായി വിജയൻ). The individual's name is "Vijayan" (വിജയൻ), while "Pinarayi" (പിണറായി) is the place where he is from. The title list would consist of both "പിണറായി-Place" and "പിണറായി വിജയൻ- Person". When a bigram check is employed first, all instances of "പിണറായി വിജയൻ" are annotated with the tag "Person". The tokens in the article body are annotated "പിണറായി- Person, വിജയൻ- Person". If this is followed by the annotation of "പിണറായി", the token "പിണറായി വിജയൻ" is modified to " പിണറായി- Place, വിജയൻ-Person". To avoid this behaviour for Malayalam, uni-grams are always annotated first and then followed by higher order n grams.

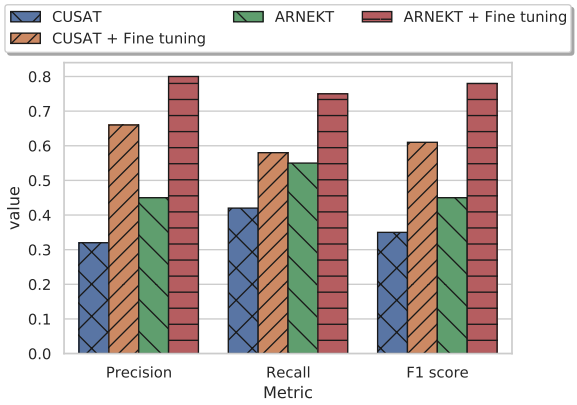
Punctuation in names Another common practice is the usage of acronyms within the name. For example, *Madath Thekkepaattu Vasudevan Nair* usually goes by *M.T. Vasudevan Nair* (എം.ടി. വാസുദേവൻ നായർ). The name is sometimes tokenized as ("എം.", "ടി.", "വാസുദേവൻ", "നായർ") or as ("എം" ,"." , "ടി.", "വാസുദേവൻ", "നായർ"). In some other cases, the article omits the punctuation, and prints the name as ("എം", "ടി", "വാസുദേവൻ", "നായർ"). Since the number of tokens within the title changes, the n-gram search consequently varies. Since this issue is specific to full stops ("."), all full stops are removed from both the article and the titles during the search phase. After all appearances of the individual tokens sans punctuation are annotated within the article, the full stops are reinserted. If the tokens to either side of the full stop have the same NE tag, the full stop is given the same tag as the tokens that wrap around it. All end-of-sentence full stops are annotated with the tag OTHER.

5.4 Language-Specific Adaptation for isiZulu

As isiZulu focuses on prefixes rather than suffixes, we perform prefix stripping for isiZulu. To this end, we make use of the capitalization described in Section 3.2. Where we could not find a full match between a title and a word in the list, we matched titles and occurrences in the text from the first capital after the initial letter. Thus, we were able to match *iGoli* and *eGoli*, i.e. Johannesburg



(a) Variation of F1 score with the amount of test data used for fine tuning



(b) Variation in model performance after fine tuning

Figure 3: Fine tuning analysis

→ from/in Johannesburg.

6 Experiments and Machine Learning Setup

For comparison, we use two baseline systems. One rule-based baseline annotation system and a neural network baseline. The rule-based baseline directly annotates the data sets with the title list generated in section 5.1. A bi-gram search is used to annotate titles that have two words. This procedure does not account for inflections and annotates perfect matches in the corpus. The rule-based baseline is therefore language independent. This system is used to evaluate the importance of accommodating inflection and agglutination when compiling an NER data set for morphologically complex languages.

The deep learning baseline for NER is implemented using Keras (Chollet, 2015). It is a recurrent LSTM network with the following layers:

1. Trainable linear embeddings of size 200
2. Bidirectional LSTM with 45 units for each direction; recurrent dropout probability of 0.1
3. Linear layer with 50 units and ReLU activation, applied to each time step
4. CRF layer with four units (one per NE class)

The model is trained using the RMSprop optimizer with a learning rate of 0.001 for 10 epochs.

We use XLM-RoBERTa (Conneau et al., 2019) to build the NER system. It is a pre-trained multilingual transformer model which has successfully been applied to low resourced languages such as Swahili and Urdu. The model is trained in the `xlm-roberta-base` configuration using decoupled weight decay (Loshchilov and Hutter, 2019) and layer-wise decaying learning rates (Sun et al., 2019). The embedding layer is frozen to avoid overfitting. We train the model on a TPU in Google Colab[8] using `bfloat16` mixed precision training and the following hyperparameters:

- Sequence length: 50
- Batch size: 1024
- Epochs: 10
- Base learning rate: $2 \cdot 10^{-5}$
- Weight decay factor: 0.99
- Learning rate decay factor: 0.95

The data sets are split into training, testing and validation sets by a 80:10:10 ratio.

6.1 Fine Tuning

Before testing the model with a target data set, the model is fine tuned for adaptation. A small subset of each test set is used to tune the weights and the remaining data is used to test the model. Fine tuning is carried out for two reasons: (i) to accommodate for changes in writing style and format and (ii) to expose the model to previously unseen tokens. Since agglutination and heavy inflection exists in both languages, it is practically infeasible to construct dictionaries that account for all words

[8] https://colab.research.google.com/

Table 2: XLM-RoBERTa results for Malayalam

	WikiMl			CUSAT			ARNEKT		
Class	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Person	0.94	0.80	0.87	0.65	0.48	0.56	0.74	0.65	0.69
Place	0.93	0.83	0.87	0.55	0.57	0.56	0.76	0.78	0.77
Organization	0.79	0.82	0.81	0.48	0.28	0.35	0.75	0.62	0.68
Other	0.99	1.00	0.99	0.99	0.99	0.99	0.96	0.97	0.97
Macro Average	**0.94**	**0.80**	**0.87**	**0.66**	**0.58**	**0.61**	**0.80**	**0.75**	**0.78**

Table 3: XLM-RoBERTa results for isiZulu

	WikiZu			NCHLT II		
Class	Precision	Recall	F1 Score	Precision	Recall	F1 Score
Place	0.94	0.87	0.90	0.46	0.41	0.43
Person	0.78	0.90	0.84	0.31	0.20	0.24
Organization	0.78	0.75	0.77	0.25	0.15	0.19
Other	0.99	0.99	0.99	0.95	0.97	0.96
Macro Average	**0.9**	**0.88**	**0.89**	**0.69**	**0.56**	**0.63**

in them. During the training phase, a dictionary is created using the developed data set which is then used to feed tokens into an embedding layer. In the case of an external data set, the model encounters many words foreign to its dictionary. Fine tuning helps it to learn the appearance patterns of unknown tokens within the article.

6.2 Results for Malayalam

Figure 3a depicts the model performance when varying amounts of test data are used for fine tuning. For the ARNEKT data, tuning the model with a small portion of the test data set increases the performance drastically. Since this data set is large, even a small portion of it helps the model adapt easily. On the other hand, the smaller CUSAT data set attains a noticeable increase in model performance at a slightly higher level of fine tuning. Since fine tuning requires a sufficient amount of tokens, a slightly bigger chunk of the CUSAT data set has to be used to fine-tune the model's parameters. The effect of fine tuning on the overall performance is visualised in Figure 3b (with 10% data for ARNEKT and 20% for CUSAT). The model performance increases considerably after fine tuning, in both cases. The class-wise performance for the WikiML, ARNEKT and CUSAT data sets is shown in Table 2. As expected, the (upper bound) results for WikiML are high across all NE classes. For the ARNEKT data set, the model performs also quite well with an average F1 score of 0.78. In comparison, the out-of-domain evaluation on the CUSAT data obtains an F1 score of 0.61, with particularly low results for *org* entities. This may be due to the fact that organisations are distributed differently in this domain.

The baseline annotation system was also evaluated on the CUSAT data set and the ARNEKT data set, obtaining F1-scores of 0.29 and 0.39, respectively. This performance highlights the importance of considering inflections, and fine tuning the model for domain adaptation.

6.3 Results for isiZulu

Table 3 shows the results of porting our system to isiZulu (with 20% of the data for fine-tuning). With an average F1-Score of 0.87 our system performs well on the in-domain WikiZu data but worse in the out-of-domain evaluation on NHCLT II, with an average F1-Score of 0.45. It still easily outperforms the rule-based baseline system (0.24 F1-Score) and the LSTM baseline (0.45 F1-Score). The lower performance compared to Malayalam can be explained by the fact that the domain of the test set is very different from that of the training set (legal vs. Wikipedia) and, moreover, the training set for isiZulu is considerably smaller than for Malayalam. In this context and given that we only used 2 days to tweak the system for isiZulu, we still consider the results an encouraging first step.

7 Analysis of Results

Since testing on WikiML can be regarded as in-domain, we focus on the analysis of errors on the

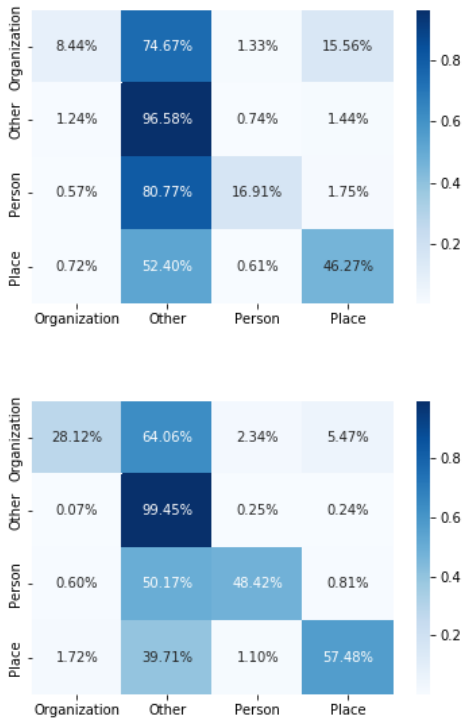


Figure 4: Confusion Matrix for CUSAT before (above) and after (below) fine tuning

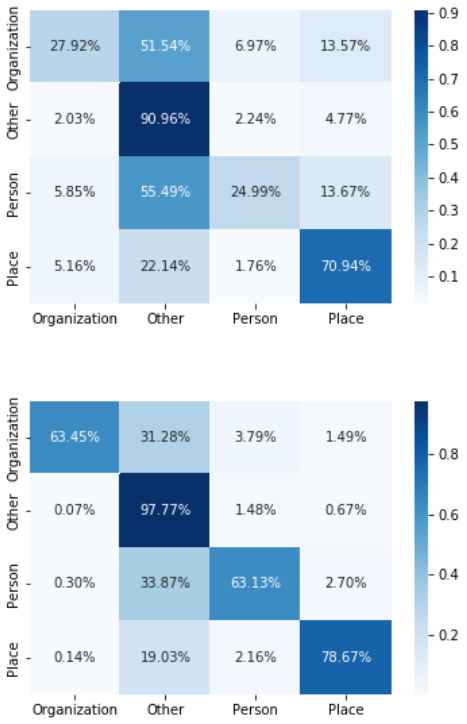


Figure 5: Confusion Matrix for ARNEKT before (above) and after (below) fine tuning

external data sets. Figures 4 and 5 display the confusion matrices of the results obtained when testing our model before and after fine tuning. In both cases, the biggest source of error is observed to be misclassification into the "Other" category. This is expected in languages with morphological complexities, since named entities are concealed within agglutinations and suffixes. It should be noted that for ARNEKT, the performance errors do not necessarily originate from the model. To our knowledge, ARNEKT was not manually annotated, but created with rule based annotation procedures and word lists. Consequently, annotation errors can be observed within the data set. In some cases, the WikiML model is seen to predict correct named entity tags for tokens wrongly annotated in ARNEKT. Two examples are presented in (4) and (5). Wrong annotations have been highlighted in red. Fine tuning clearly improves the impact of errors involving the "Other" category significantly.

(4) Tokens: ഡച്ച് സാമ്പത്തിക
ARNEKT: Place Other
Prediction: Organization Other
ശാസ്ത്രജ്ഞൻ ആണ് യാൻ ടിൻബർജെൻ
Other Other Other Other
Other Other Person Person

Jan Tinbergen is a Dutch Economist

(5) Tokens: ശ്രീലങ്കൻ ക്രിക്കറ്റ് ചരിത്രത്തിലെ
ARNEKT: Other Other Person
Prediction: Place Person Other
ഏറ്റവും മികച്ച താരങ്ങളിലൊരാളാണ് മുരളി
Other Other Other Person
Other Other Other Person

Murali is one of the best players in the history of Sri Lankan cricket

For CUSAT, the presence of out-of-domain/unseen words is clearly the cause of most errors in the vanilla model. Once fine-tuned with a portion of the data set, this is reduced significantly.

Disregarding the "Other" class, the model seems to confuse "Person" and "Organization" entites with "Place" entities in both data sets. This is almost always observed with multi worded entities that have places embedded in their names. Cases include people with places attached to them (as explained in section 5.3.1) and organizations with the same characteristic (e.g. "New York Public Library"). The "New York" portion in such entities can be thought of as a place entity embedded in

an organization entity, or can be viewed simply as an organization entity without taking into account embedded entity classes. For one-worded entities, errors can often be seen to arise from annotation variations between the ground truth and the automatically generated dataset. For example, words such as “Library” and “College” are mapped as ”Place” entities by the Google Knowledge graph during the generation of title lists. Subsequently, instances of such words are labeled as ”Place” by our vanilla model trained on the WikiML dataset. However, the external datasets label them as ”Organization” entities in some cases, which indirectly translates to mistakes during evaluation.

8 Conclusion

We demonstrated the implementation of a fully automated pipeline for the creation of a named entity tagged data set with freely available resources. We showed how the pipeline can be adaptated for two morphologically complex, agglutinating languages. Finally, we propose an easily portable, weakly supervised NER system for Malayalam and isiZulu based on this pipeline. The system can be developed quickly: We spent 2 weeks on developing the initial system for Malayalam and 2 days for porting it to isiZulu. We tested in- and out-of-domain on a number of publicly available data sets, with encouraging results, especially for Malayalam.

References

A.P Ajees and Sumam Mary Idicula. 2018. A named entity recognition system for Malayalam using neural networks. *Procedia Computer Science*, 143:962 – 969. 8th International Conference on Advances in Computing & Communications (ICACC-2018).

K. K. Akhil, R. Rajimol, and V. S. Anoop. 2020. Parts-of-speech tagging for Malayalam using deep learning techniques. *International Journal of Information Technology*.

Mahathi Bhagavatula, Santosh GSK, and Vasudeva Varma. 2012. Language independent named entity identification using Wikipedia. In *Proceedings of the First Workshop on Multilingual Modeling*, pages 11–17, Jeju, Republic of Korea. Association for Computational Linguistics.

Premjith Bhavukam, Soman K.P., and M Anand Kumar. 2018. A deep learning approach for Malayalam morphological analysis at character level. *Procedia Computer Science*, 132:47 – 54. International Conference on Computational Intelligence and Data Science.

MS Bindu and Sumam Mary Idicula. 2011. Named entity identifier for Malayalam using linguistic principles employing statistical methods. *International Journal of Computer Science Issues(IJCSI)*, 8(5):185–191.

Dhouha Bouamor, Adrian Popescu, Nasredine Semmar, and Pierre Zweigenbaum. 2013. Building specialized bilingual lexicons using large scale background knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a dataset for named entity recognition in the archaeology domain. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4573–4577, Marseille, France. European Language Resources Association.

Thorsten Brants. 2002. TnT: A statistical part-of-speech tagger. *ANLP*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

François Chollet. 2015. Keras. `https://keras.io`.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

G. Remmiya Devi, P.V. Veena, M. Anand Kumar, and K.P. Soman. 2016. Entity extraction for Malayalam social media text using structured skip-gram based embedding features from unlabeled data. *Procedia Computer Science*, 93:547 – 553. Proceedings of the 6th International Conference on Advances in Computing and Communications.

Roald Eiselen. 2016. Government domain named entity recognition for South African languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3344–3348.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Barathi Ganesh Hullathy Balakrishnan, Soman KP, Reshma U, Mandar Kale, Prachi Mankame, Gouri Kulkarni, Anitha Kale, and Anand Kumar M. 2018. Information extraction for conversational systems in Indian languages - Arnekt IECSIL. In *Proceedings of the 10th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE'18, page 18–20, New York, NY, USA. Association for Computing Machinery.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic. Association for Computational Linguistics.

Steven Krauwer. 2003. The Basic Language Resource Kit (BLARK) as the first milestone for the language resources roadmap. In *Proceedings of the 2003 International Workshop Speech and Computer (SPECOM 2003)*, pages 8–15.

S. Kumar, M. Anand Kumar, and K.P. Soman. 2019. Deep learning based part-of-speech tagging for Malayalam Twitter data (special issue: Deep learning techniques for natural language processing). *Journal of Intelligent Systems*, 28(3):423 – 435.

Patrick Littell, Kartik Goyal, David R. Mortensen, Alexa Little, Chris Dyer, and Lori Levin. 2016. Named entity recognition for linguistic rapid response in low-resource languages: Sorani Kurdish and Tajik. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 998–1006, Osaka, Japan. The COLING 2016 Organizing Committee.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545, Copenhagen, Denmark. Association for Computational Linguistics.

S. K. Nambiar, A. Leons, S. Jose, and Arunsree. 2019. POS tagger for Malayalam using Hidden Markov Model. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 957–960.

Jian Ni and Radu Florian. 2016. Improving multilingual named entity recognition with Wikipedia entity type mapping. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1275–1284, Austin, Texas. Association for Computational Linguistics.

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming Wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132, Hobart, Australia.

Joel Nothman, Tara Murphy, and James R. Curran. 2009. Analysing Wikipedia and gold-standard corpora for NER training. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 612–620, Athens, Greece. Association for Computational Linguistics.

Jisha P Jayan, Rajeev R R, and Elizabeth Sherly. 2013. A hybrid statistical approach for named entity recognition for Malayalam language. In *Proceedings of the 11th Workshop on Asian Language Resources*, pages 58–63, Nagoya, Japan. Asian Federation of Natural Language Processing.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Will Radford, Xavier Carreras, and James Henderson. 2015. Named entity recognition with document-specific KB tag gazetteers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 512–517, Lisbon, Portugal. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Alexander E. Richman and Patrick Schone. 2008. Mining Wiki resources for multilingual named entity recognition. In *Proceedings of ACL-08: HLT*, pages 1–9, Columbus, Ohio. Association for Computational Linguistics.

S. Shruthi, Jiljo, and P.V. Pranav. 2016. A study on named entity recognition for Malayalam language using TnT tagger & maximum entropy Markov model. *International Journal of Applied Engineering Research*, 11:5425–5429.

Amit Singhal. 2012. Introducing the Knowledge Graph: things, not strings. https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

P S Sreeja and Anitha S Pillai. 2020. Towards an efficient Malayalam named entity recognizer analysis on the challenges. *Procedia Computer Science*, 171:2541 – 2546. Third International Conference on Computing and Network Communications (CoCoNet'19).

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Elsabé Taljard and Sonja E. Bosch. 2006. A comparison of approaches to word class tagging: Disjunctively vs. conjunctively written Bantu languages. *Nordic journal of African studies*, 15(4):428–442.

Antonio Toral and Rafael Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*.