calamanCy: A Tagalog Natural Language Processing Toolkit

Lester James V. Miranda
ExplosionAI GmbH
lj@explosion.ai

Abstract

Despite the presence of language resources for Tagalog, there is still no unifying framework to apply them to downstream tasks. We introduce calamanCy, an open-source toolkit for constructing natural language processing (NLP) pipelines. It is built on top of spaCy, enabling easy experimentation and integration with other frameworks. calamanCy addresses the development gap by providing a consistent API for building NLP applications and offering efficient multitask models for core NLP tasks. Additionally, we include structured evaluations for dependency parsing, part-of-speech (POS) tagging, and named entity recognition (NER). calamanCy aims to accelerate the progress of Tagalog NLP by consolidating disjointed resources in a unified framework. The calamanCy Github repository can be found at https://github.com/ljvmiranda921/calamanCy.

1 Introduction

Tagalog is a low-resource language belonging to the Austronesian family, with over 76 million speakers in the Philippines (Lewis, 2009). Despite its speaker population, there are limited resources available for the language (Cruz and Cheng, 2021). Nevertheless, pretrained language models (LMs) for Tagalog exist. These models include multilingual LMs such as XLM-R and mBERT (Conneau et al., 2019; Devlin et al., 2019), as well as monolingual LMs such as RoBERTa Tagalog (Cruz and Cheng, 2021). On the data side, language resources are dispersed. Tagalog treebanks exist within the Universal Dependencies framework (Dehouck and Denis, 2019; Kondratyuk, 2019; Aquino and de Leon, 2020), while domain-specific corpora in literature are scarce (Enriquez and Estuar, 2023; Livelo and Cheng, 2018). Within this context, a coherent and unified framework for all core NLP tasks in Tagalog is still lacking. This issue hampers model development, experimental workflows, and the overall advancement of Tagalog NLP.

To address these challenges, we introduce calamanCy,[1] an open-source toolkit for Tagalog NLP. It is built on top of spaCy (Honnibal et al., 2020) and offers end-to-end pipelines for NLP tasks, including dependency parsing, parts-of-speech (POS) tagging, and named entity recognition (NER). calamanCy also provides models of different sizes, with a balance between performance and accuracy. Finally, our work has two main contributions: (1) a unified framework via calamanCy, and (2) structured benchmarks for Tagalog NLP tasks using efficient multitask pipelines.

2 Related Work

Open-source toolkits for NLP There has been a growing body of work in the development of NLP toolkits tailored to specific settings in recent years. For languages, these software include DaCy for Danish (Enevoldsen et al., 2021) and HuSpaCy for Hungarian (Orosz et al., 2022). For domains, there is medspaCy for clinical text (Eyre et al., 2021) and scispaCy for scientific text (Neumann et al., 2019). These tools were based on spaCy (Honnibal et al., 2020), an industrial-strength open-source software for natural language processing. Using spaCy as a foundation to build NLP toolkits is an optimal choice given its popularity and integration with other frameworks such as HuggingFace (Wolf et al., 2019). However, no tool exists for Tagalog until now. In this paper, we will showcase how calamanCy provides similar capabilities as DaCy and HuSpaCy using Tagalog resources.

Evaluations on Tagalog NLP Tasks Structured evaluations for core NLP tasks, such as dependency parsing, POS tagging, and NER, are sparse. However, we have access to a reasonable amount of

[1] The name “calamanCy” came from *kalamansi*, a citrus fruit native to the Philippines.

Entity	Description	Examples
Person (PER)	Person entities limited to humans. It may be a single individual or group.	Juan de la Cruz, Jose Rizal, Quijano de Manila
Organization (ORG)	Organization entities limited to corporations, agencies, and other groups of people defined by an organizational structure.	Meralco, DPWH, United Nations
Location (LOC)	Location entities are geographical regions, areas, and landmasses. Geo-political entities are also included within this group.	Pilipinas, Manila, CALABARZON, Ilog Pasig

Table 1: Entity types used for annotating `calamanCy-gold` (derived from the TLUnified corpus of Cruz and Cheng, 2021). Annotation guidelines can be found at `https://github.com/ljvmiranda921/calamanCy/tree/master/datasets/tl_calamancy_gold_corpus/guidelines`

data to conduct comprehensive benchmarks. For example, TLUnified (Cruz and Cheng, 2021) is a pretraining corpus that combines news reports (Cruz et al., 2020), a preprocessed version of CommonCrawl (Suarez et al., 2019), and several other datasets. However, it was evaluated on domain-specific applications that may not easily transfer to more general tasks. For dependency parsing and POS tagging, we have Universal Dependencies treebanks such as TRG (Dehouck and Denis, 2019; Kondratyuk, 2019) and Ugnayan (Aquino and de Leon, 2020). This paper will fill the evaluation gap by providing structured benchmarks on core NLP tasks.

3 Implementation

The best way to use calamanCy is through its trained pipelines. After installing the library, users can access the models via:

```
import calamancy as cl
nlp = cl.load("tl_calamancy_md-0.1.0")
```

Here, the variable `nlp` is a spaCy processsing pipeline.[2] It contains trained components for POS tagging, dependency parsing, and NER. calamanCy offers three pipelines of varying capacity: two word vector-based models (`md`, `lg`), and one transformer-based model (`trf`). We will discuss how these pipelines were developed in the following section.

3.1 Pipeline development

Data annotation To construct the NER corpus, we curated a portion of TLUnified (Cruz and Cheng, 2021) to only contain Tagalog news articles. Including the author, we recruited two more annotators who have at least a Bachelors degree and whose native language is Tagalog. The three annotators labeled over the course of four months given three entity types as seen in Table 1. The entity types were chosen to resemble ConLL (Sang, 2002; Sang and Meulder, 2003), a standard NER benchmark. We measured inter-annotator agreement (IAA) by taking the pairwise Cohen's κ without the un-annotated tokens (as recommended by Deléger et al., 2012) then averaged them for all three pairs. This process resulted to a Cohen's κ score of 0.78. To avoid confusing with the original TLUnified corpora, we will refer to this annotated NER dataset as `calamanCy-gold`. The final dataset statistics can be found in Table 2.

Dataset	Examples	PER	ORG	LOC
Training	6252	6418	3121	3296
Development	782	793	392	409
Test	782	818	423	438

Table 2: Dataset statistics for `calamanCy-gold`.

Model training We considered three design dimensions when training the calamanCy pipelines: (1) presence of pretraining, (2) the word representation, and (3) the representation size. *Pretraining* involves learning vectors from raw text to better inform model initialization. This process is done using a variant of the cloze task (Devlin et al., 2019). Here, the pretraining objective asks the model to predict some number of leading and training UTF-8 bytes for the words. *Word representations* may either involve training static vectors using floret,[3] an efficient version of fastText (Bojanowski et al.,

[2] `https://spacy.io/usage/processing-pipelines`

[3] `https://github.com/explosion/floret`

2016), or using context-sensitive vectors from a transformer (Vaswani et al., 2017). Finally, *representation size* is an engineering dimension determined via a performance-accuracy tradeoff.

The general process involves pretraining a filtered version of TLUnified (removing overlaps with `calamanCy-gold`), constructing static vectors if necessary, and training the downstream components. We trained the NER component using data from `calamanCy-gold`, while the dependency parser and POS tagger were trained using the Ugnayan treebank. In the end, we came up with three language pipelines of varying sizes:[4]

- `tl_calamancy_md`: Medium-sized Tagalog pipeline optimized for CPU. Pretrained using raw texts from TLUnified. Includes a static floret vector table containing 50k unique vectors (200 dimensions).
- `tl_calamancy_lg`: Large-sized Tagalog pipeline optimized for CPU. Same training setup as the medium-sized model. The only difference is this pipeline contains 200k unique vectors (200 dimensions).
- `tl_calamancy_trf`: Tagalog transformer pipeline. No pretraining. Instead of a static vector table, it uses context-sensitive vectors from the roberta-tagalog-base transformer (Cruz and Cheng, 2021).

The full training configuration for v0.1.0 of the calamanCy pipelines can be found on Github: `https://github.com/ljvmiranda921/calamanCy/tree/master/models/v0.1.0`.

4 Evaluation

5 Discussions

6 Conclusion

References

Angelina A. Aquino and Franz A. de Leon. 2020. Parsing in the absence of related languages: Evaluating low-resource dependency parsers on Tagalog. In *Universal Dependencies Workshop*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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. In *Annual Meeting of the Association for Computational Linguistics*.

Jan Christian Blaise Cruz and Charibeth Ko Cheng. 2021. Improving Large-scale Language Models and Resources for Filipino. In *International Conference on Language Resources and Evaluation*.

Jan Christian Blaise Cruz, Jose Kristian Resabal, James Lin, Dan John Velasco, and Charibeth Ko Cheng. 2020. Exploiting News Article Structure for Automatic Corpus Generation of Entailment Datasets. In *Pacific Rim International Conference on Artificial Intelligence*.

Mathieu Dehouck and P. Denis. 2019. Phylogenic Multi-Lingual Dependency Parsing. In *North American Chapter of the Association for Computational Linguistics*.

Louise Deléger, Qi Li, Todd Lingren, Megan Kaiser, Katalin Molnár, Laura Stoutenborough, Michal Kouril, Keith A. Marsolo, and Imre Solti. 2012. Building Gold Standard Corpora for Medical Natural Language Processing Tasks. *Proceedings of the AMIA Annual Symposium*, 2012:144–53.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, abs/1810.04805.

Kenneth C. Enevoldsen, L M Hansen, and Kristoffer Laigaard Nielbo. 2021. DaCy: A Unified Framework for Danish NLP. In *Workshop on Computational Humanities Research*.

Raphael Christen K. Enriquez and Maria Regina Justina Estuar. 2023. Determining Linguistic Features of Hate Speech from 2016 Philippine Election-Related Tweets. *2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD)*, pages 1–6.

Hannah Eyre, Alec B. Chapman, Kelly S. Peterson, Jianlin Shi, Patrick R. Alba, Makoto M. Jones, Tamára L Box, Scott L Duvall, and Olga V Patterson. 2021. Launching into clinical space with medspaCy: a new clinical text processing toolkit in Python. *Proceedings of the AMIA Annual Symposium*, 2021:438–447.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

D. Kondratyuk. 2019. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. In *Conference on Empirical Methods in Natural Language Processing*.

[4] The naming convention resembles spaCy's model names: `{language code}_{source}_{size}`

Paul M. A. Lewis. 2009. Ethnologue: languages of the world. https://ethnologue.com/language/tgl. Accessed: June 2023.

Evan Dennison S. Livelo and Charibeth Ko Cheng. 2018. Intelligent Dengue Infoveillance Using Gated Recurrent Neural Learning and Cross-Label Frequencies. *2018 IEEE International Conference on Agents (ICA)*, pages 2–7.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. *ArXiv*, abs/1902.07669.

György Orosz, Zsolt Szántó, Péter Berkecz, Gergo Szabó, and Richárd Farkas. 2022. HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. *ArXiv*, abs/2201.01956.

Erik Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *ArXiv*, cs.CL/0209010.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *ArXiv*, cs.CL/0306050.

Pedro Ortiz Suarez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Conference on Neural Information Processing Systems*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.