calamanCy: A Tagalog Natural Language Processing Toolkit

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

Abstract

We introduce calamanCy, an open-source toolkit for constructing natural language processing (NLP) pipelines for Tagalog. 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 general-purpose multitask models with out-of-the-box support for dependency parsing, parts-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 toolkit is available on GitHub: https://github.com/ljvmiranda921/calamanCy.

1 Introduction

Tagalog is a low-resource language from the Austronesian family, with over 28 million speakers in the Philippines (Lewis, 2009). Despite its speaker population, few resources exist for the language (Cruz and Cheng, 2021). For example, Universal Dependencies (UD) treebanks for Tagalog are tiny ($\ll$ 20k words) (Samson, 2018; Aquino and de Leon, 2020), while domain-specific corpora are sparse (Cabasag et al., 2019; Livelo and Cheng, 2018). In addition, Tagalog language models (LMs) (Cruz and Cheng, 2021; Jiang et al., 2021) are few, while most multilingual LMs (Conneau et al., 2019; Devlin et al., 2019) underrepresent the language (Lauscher et al., 2020). Thus, consolidating these disjointed resources in a coherent framework is still an open problem. The lack of such framework hampers model development, experimental workflows, and the overall advancement of Tagalog NLP.

To address this problem, 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 such as dependency parsing, parts-of-speech (POS) tagging, and named entity recognition (NER). calamanCy also provides models of different sizes to fit any performance or accuracy requirements. This work has two main contributions: (1) an open-source toolkit containing general-purpose multitask pipelines with out-of-the box support for common NLP tasks, and (2) comperehensive benchmarks on several Tagalog core NLP tasks.

2 Related Work

Open-source toolkits for NLP There has been a growing body of work in developing NLP toolkits in recent years. For languages, these include DaCy for Danish (Enevoldsen et al., 2021) and HuSpaCy for Hungarian (Orosz et al., 2022). For domain-specific data, there is medspaCy for clinical text (Eyre et al., 2021) and scispaCy for scientific documents (Neumann et al., 2019). These tools employ spaCy (Honnibal et al., 2020), an industrial-strength open-source software for natural language processing. Using spaCy as a foundation is optimal, given its popularity and tight integration with other frameworks like HuggingFace transformers (Wolf et al., 2019). However, no tool has existed for Tagalog until now. We aim to fill this development gap through calamanCy.

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 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. In addition, Tagalog has

[1] "calamanCy" derives its name 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 TLUnified-NER (derived from the TLUnified pretraining corpus of Cruz and Cheng, 2021).

two Universal Dependencies treebanks, Tagalog Reference Grammar (TRG) (Samson, 2018) and Ugnayan (Aquino and de Leon, 2020), with POS tags and relational structures. This paper will fill the evaluation gap by providing structured benchmarks on these core 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 processing pipeline.[2] It contains trained components for POS tagging, dependency parsing, and NER. calamanCy offers three pipelines of varying capacity: two static word vector-based models (md, lg), and one transformer-based model (trf). We will discuss how we developed these pipelines in the following section.

3.1 Pipeline development

Data annotation for NER There is no gold-standard corpus for NER, so we built one. To construct the NER corpus, we curated a portion of TLUnified (Cruz and Cheng, 2021) to contain Tagalog news articles. Including the author, we recruited two more annotators with at least a bachelor's degree and whose native language is Tagalog. The three annotators labeled for four months, given three entity types as seen in Table 1. We chose the entity types to resemble ConLL (Sang, 2002; Sang and Meulder, 2003), a standard NER benchmark. We measured inter-annotator agreement (IAA) by

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 TLUnified-NER.

taking the pairwise Cohen's κ on all tokens and then averaged them for all three pairs. This process resulted in a Cohen's κ score of 0.81. To avoid confusion with the original TLUnified pretraining corpora, we will refer to this annotated NER dataset as TLUnified-NER. The final dataset statistics can be found in Table 2. For the dependency parser and POS tagger, we merged the TRG (Samson, 2018) and Ugnayan (Aquino and de Leon, 2020) treebanks to leverage their small yet relevant examples.

Model training We considered three design dimensions when training the calamanCy pipelines: (1) the presence of pretraining, (2) the word representation, and its (3) size or dimension. Model *pretraining* involves learning vectors from raw text to inform its initialization better. Here, the pretraining objective asks the model to predict some number of leading and trailing UTF-8 bytes for the words—a variant of the cloze task (Devlin et al., 2019). A model's *word representation* may involve training static word embeddings using floret,[3] an efficient version of fastText (Bojanowski et al., 2016), or using context-sensitive vectors from a transformer (Vaswani et al., 2017). Finally, a model's *dimension* is our way to tune the tradeoff between performance and accuracy.

The general process involves pretraining a fil-

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

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

Pipeline	Pretraining objective	Word embeddings	Dimensions
Medium-sized pipeline (tl_calamancy_md)	Predict some number of leading and trailing UTF-8 bytes for the words.	Uses floret vectors trained on the TLUnified corpora.	50k unique vectors (200 dimensions), Size: 77 MB
Large-sized pipeline (tl_calamancy_lg)	Same pretraining objective as the medium-sized pipeline.	Uses fastText vectors trained on CommonCrawl corpora.	714k unique vectors (300 dimensions), Size: 455 MB
Transformer-based pipeline (tl_calamancy_trf)	No separate pretraining because there's no token-to-vector component.	Context-sensitive vectors from a transformer network.	Uses roberta-tagalog-base. Size: 813 MB

Table 3: Language pipelines available in calamanCy (v0.1.0). The pretraining method for the word-vector models is a variant of the *cloze task*. All pipelines have a `tagger`, `parser`, `morphologizer`, and `ner` spaCy component.

Dataset	Task / Labels	Description
Hatespeech (Cabasag et al., 2019)	Binary text classification *(hate speech, not hate speech)*	Contains 10k tweets collected during the 2016 Philippine Presidential Elections labeled as hate speech or non-hate speech.
Dengue (Livelo and Cheng, 2018)	Multilabel text classification *(absent, dengue, health, sick, mosquito)*	Contains 4k dengue-related tweets collected for a health infoveillance application that classifies text into dengue subtopics.
TLUnified-NER (Cruz and Cheng, 2021)	Named entity recognition *(Person, Organization, Location)*	A held-out test split from the annotated TLUnified corpora containing news reports and other articles. See Table 2.
Merged UD (Samson, 2018; Aquino and de Leon, 2020)	Dependency parsing and POS tagging	Merged version of the Ugnayan and TRG treebanks from the Universal Dependencies framework.

Table 4: Datasets for benchmarking calamanCy.

tered version of TLUnified, constructing static word embeddings if necessary, and training the downstream components. We used TLUnified-NER to train the NER component and trained the dependency parser and POS tagger using the combined treebanks. Ultimately, we devised three language pipelines as seen in Table 3.

4 Evaluation

Architectures We used spaCy's built-in architectures for each component in the calamanCy pipeline. The token-to-vector layer uses the multi-hash embedding trick (Miranda et al., 2022) to reduce the representation size. For the parser and named entity recognizer, we used a transition-based parser that maps text representations into a series of state transitions. As for the text categorizer, we utilized an ensemble of a bag-of-words model and a feed-forward network.

Experimental set-up We assessed the calamanCy pipelines on various Tagalog benchmarks as detailed in Table 4. We also tested on text categorization, an unseen task, for robustness. For NER evaluation, we used a held-out test split from TLUnified-NER. However, for dependency parsing and POS tagging, the combined treebank is still small ($\ll$ 20k tokens), so we followed the Universal Dependencies data split guidelines and performed 10-fold cross-validation (Nivre et al., 2022). For all the other datasets, we computed their performance across five trials and then reported the average and standard deviation.

We also tested a cross-lingual transfer learning approach, i.e., finetuning a model from a source language closely related to Tagalog. According to Aquino and de Leon (2020), the closest languages to Tagalog are Indonesian (id), Ukrainian (uk), Vietnamese (vi), Romanian (ro), and Catalan (ca). They used a metric (Agić, 2017) based on the

Model	Text categorization Hatespeech (binary)	Dengue (multilabel)	NER TLUnified-NER	Dep. pars. & POS tag. Merged UD, UAS / LAS	Merged UD, POS Acc.
Monolingual (Ours)					
tl_calamancy_md	74.40±0.05	65.32±0.04	87.67±0.03	76.47 / 54.40	96.70
tl_calamancy_lg	75.62±0.02	68.42±0.01	88.90±0.01	82.13 / 70.32	97.20
tl_calamancy_trf	**78.25±0.06**	**72.45±0.02**	**90.34±0.02**	**92.48 / 80.90**	**97.80**
Cross-lingual transfer					
uk_core_news_trf	75.24±0.03	65.57±0.01	51.11±0.02	54.77 / 37.68	82.86
ro_core_news_lg	69.01±0.01	59.10±0.01	02.01±0.00	84.65 / 65.30	82.80
ca_core_news_trf	70.01±0.02	59.42±0.03	14.58±0.02	91.17 / 79.30	83.09
Multilingual finetuning					
xlm-roberta-base	77.57±0.01	67.20±0.01	88.03±0.03	88.34 / 76.07	94.29
bert-base-multilingual	76.40±0.02	71.07±0.04	87.40±0.02	90.79 / 78.52	95.30

Table 5: Benchmark evaluation scores for monolingual, cross-lingual, and multilingual pipelines across a variety of tasks and datasets. We evaluated the text categorization and NER tasks across five trials, and then conducted 10-fold cross-validation for dependency parsing. F1-scores are reported on the text categorization and NER tasks.

World Atlas for Language Structures (Haspelmath et al., 2005). However, only uk, ro, and ca have equivalent spaCy pipelines, so we only compared against those three. Finally, we also compared against multilingual language models. We fine-tuned our datasets on XLM RoBERTa (Conneau et al., 2019) and an uncased version of multilingual BERT (Devlin et al., 2019). These LMs contain Tagalog in their training pool and are common alternatives for building Tagalog NLP applications.

5 Discussion

Table 5 shows the F1-scores for the text categorization and NER tasks and the unlabeled (UAS) and labeled attachment scores (LAS) for the dependency parsing task.

The calamanCy pipelines are competitive across all core NLP tasks while maintaining a smaller compute footprint. As shown in the text categorization and NER results, users with low compute budgets can attain similar performance to multilingual LMs by using medium- or large-sized calamanCy models. The transformer-based calamanCy pipeline is the best option for users who prioritize accuracy. However, we were surprised that most alternative approaches perform better in dependency parsing. We attribute this performance to the added strength of multilingual and cross-lingual information, which we don't have when training solely on a smaller treebank. We plan to improve dependency parsing performance by building a larger treebank within the Universal Dependencies framework. For practical applications, we recommend users to start with a medium- or large-sized calamanCy model before trying out GPU-intensive pipelines. Only then can they switch to a transformer-based pipeline to get accuracy gains.

6 Conclusion

In this paper, we introduced calamanCy, a natural language processing toolkit for Tagalog. Our work has two main contributions: (1) an open-source toolkit containing general-purpose multitask pipelines with out-of-the-box support for common NLP tasks, and (2) comprehensive benchmarks that compare against alternative approaches, such as cross-lingual or multilingual finetuning.

We hope that calamanCy is a step forward to improving the state of Tagalog NLP. As a low-resource language, consolidating resources into a unified framework is crucial to advance research and improve collaboration. In the future, we plan to create a more fine-grained NER benchmark corpus and extend calamanCy to natural language understanding (NLU) tasks. Finally, the project is hosted on GitHub (`https://github.com/ljvmiranda921/calamanCy`) and we are happy to receive community feedback and contributions.

References

Željko Agić. 2017. Cross-Lingual Parser Selection for Low-Resource Languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, pages 1–10.

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.

Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a Dataset for Named Entity Recognition in the Archaeology Domain. In *International Conference on Language Resources and Evaluation*.

Neil Vicente P. Cabasag, Vicente Raphael C. Chan, Sean Christian Y. Lim, Mark Edward M. Gonzales, and Charibeth K. Cheng. 2019. Hate Speech in Philippine Election-Related Tweets: Automatic Detection and Classification Using Natural Language Processing. *Philippine Computing Journal Dedicated Issue on Natural Language Processing*, pages 1–14.

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*.

Louise Deleger, Qi Li, Todd Lingren, Megan Kaiser, Katalin Molnar, Laura Stoutenborough, Michal Kouril, Keith Marsolo, and Imre Solti. 2012. Building gold standard corpora for medical natural language processing tasks. In *AMIA Annual Symposium Proceedings*, pages 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*.

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.

Martin Haspelmath, Matthew Dryer, David Gil, and Comrie Bernard. 2005. The World Atlas of Language Structures. In *Oxford University Press*.

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

Shengyi Jiang, Yingwen Fu, Xiaotian Lin, and Nankai Lin. 2021. Pre-trained Language Models for Tagalog with Multi-source Data. In *Natural Language Processing and Chinese Computing*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulic, and Goran Glavas. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Conference on Empirical Methods in Natural Language Processing*.

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.

Lester James V. Miranda, Ákos Kádár, Adriane Boyd, Sofie Van Landeghem, Anders Søgaard, and Matthew Honnibal. 2022. Multi hash embeddings in spaCy. *ArXiv*, abs/2212.09255.

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

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajivc, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. 2022. Data Release Checklist - Universal Dependencies. `https://universaldependencies.org/release_checklist.html#data-split`. Accessed: June 2023.

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.

Nils Reiter. 2017. How to develop annotation guidelines. https://sharedtasksinthedh.github.io/2017/10/01/howto-annotation/. Accessed: June 2023.

Stephanie Dawn Samson. 2018. A treebank prototype of Tagalog. Bachelor's thesis, University of Tübingen, Germany.

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.

A Appendix

A.1 Reproducibility

All the experiments and models in this paper are available publicly. Readers can head over to https://github.com/ljvmiranda921/calamanCy for all related software. Note that the XLM-RoBERTa and multilingual BERT experiments may at least require a T4 or V100 GPU.

To reproduce the calamanCy models, head over to `models/v0.1.0`. To reproduce the benchmarking experiments, head over to the `report/benchmark` directory. Readers who are interested in the training set-up (e.g., hyperparameters, architectures used, etc.) can check the configuration (`.cfg`) files in the respective project's `configs/` directory.

A.2 Building the TLUnified-NER corpus

The TLUnified-NER dataset is a named entity recognition corpus containing the *Person (PER)*, *Organization (ORG)*, and *Location (LOC)* entities. It includes news articles and other texts in Tagalog from 2009 to 2020. It was based on the TLUnified pretraining corpora by (Cruz and Cheng, 2021). The author, together with two more annotators, annotated TLUnified in the course of four months. We followed the process recommended by Reiter (2017), which included resolving disagreements and updating the annotation guidelines.

To compute the inter-annotator agreement (IAA) score, we followed Brandsen et al. (2020)'s approach. We computed Cohen's κ for (1) all tokens, and (2) only annotated tokens. In addition, we also measured the (3) pairwise F1 score without the 'O' label (Deleger et al., 2012). Table 6 shows the IAA measurements while Figure 1 shows their growth after each annotation round.

Metric	IAA
Cohen's κ on all tokens	0.81
Cohen's κ on annotated tokens only	0.65
F1 score	0.91

Table 6: Inter-annotator agreement (IAA) measurements. We obtained these values by computing for the pairwise comparisons between all annotator-pairs and averaging the results.

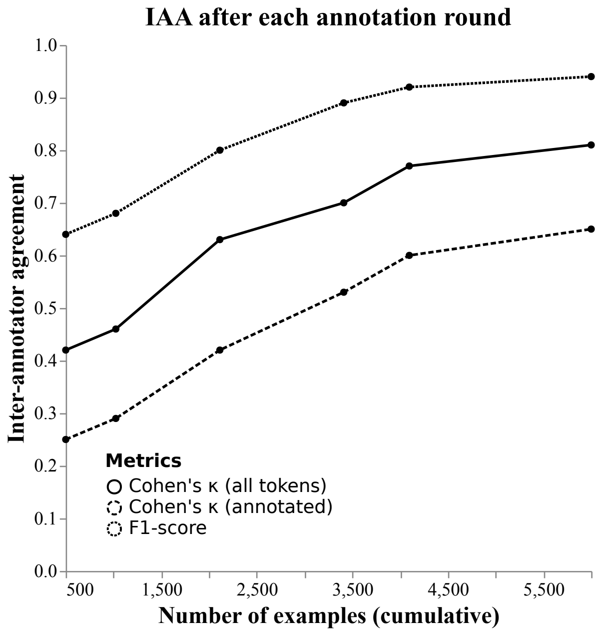

Figure 1: Inter-annotator agreement measurement after each annotation round. Each mark represents the end of a round. For each round, the annotators discuss disagreements, update the annotation guidelines, and evaluate the current set of annotations.