# Automating Dictionary Production:

# a Tagalog-English-Korean Dictionary from Scratch

# Vít Baisa<sup>1,2</sup>, Marek Blahuš<sup>1</sup>, Michal Cukr<sup>1</sup>, Ondřej Herman<sup>1,2</sup>,

Miloš Jakubíček<sup>1,2</sup>, Vojtěch Kovář<sup>1,2</sup>, Marek Medved<sup>1,2</sup>,

Michal Měchura<sup>1,2</sup>, Pavel Rychlý<sup>1,2</sup>, Vít Suchomel<sup>1,2</sup>

<sup>1</sup> Lexical Computing <sup>2</sup> Masaryk University Brno, Czech Republic {firstname.lastname}@sketchengine.eu

#### Abstract

In this paper we present lexicographic work on a Tagalog-English-Korean dictionary. The dictionary is created entirely from scratch and all of its content (besides audio pronunciation) is initially generated fully automatically from a large web corpus that we built for these purposes, and then post-edited by human editors. The full size of the dictionary is 45,000 entries, out of which 15,000 most frequent entries are manually post-edited, while the remaining 30,000 entries are left only as automated. The project is currently ongoing and will be finished in December 2019. The dictionary will be part of the online platform run by the Naver Corporation<sup>1</sup> and freely available.

**Keywords:** Sketch Engine; Lexonomy; post-editing lexicography; dictionary; corpus; Tagalog; Filipino; English; Korean

## 1. Introduction

This dictionary project is the first in the series of three, the latter two are focusing on Urdu and Lao but otherwise follow the same scheme. The goal of the project is a modern, digital, corpus-based dictionary from Tagalog (Filipino) (as source language) to English and Korean (as target languages, treated equally). The key novel aspect of the dictionary building is that the contents of the dictionary will be created fully automatically using advanced natural language processing tools and a large web corpus of Tagalog, and most of the 45,000 target entries will remain automatic. Only 15,000 most frequent entries will be post-edited. In this paper we present the dictionary as a whole and specifically focus on two major methodological issues: the automatic drafting of the dictionary and the manual post-editing.

<sup>&</sup>lt;sup>1</sup> Available at https://dict.naver.com/

# 2. On Tagalog

Tagalog is the most widely used language of the Philippines, where it is spoken by 24 million native speakers, along with additional 45 million second-language speakers who use its standardized form as the national language, officially called Filipino. It is an Austronesian language whose vocabulary has been influenced by a variety of foreign languages, most significantly English and Spanish. In spite of continuous efforts by the Philippine government to advance Filipino dating back to the 1935 constitution, and despite it being a compulsory part of the curriculum, the language is not used in all official domains; national law, business and government websites, for instance, are usually available only in English. Terminology in many fields has been reported to be inconsistent or missing, and code switching is a common practice. We have found this limiting with regard to the dictionary's coverage of certain topics.

The first comprehensive dictionary of Tagalog was compiled by Paul Klein, a Czech Jesuit missionary, in the beginning of the  $18^{\text{th}}$  century. His *Vocabulario de la lengua tagala*, inspired by earlier work by Franciscan friar Pedro de San Buenaventura, has itself become an inspiration for subsequent dictionaries of the same name, resulting in repeated reeditions until these days. Modern Tagalog is written using the Filipino Alphabet, which includes all the 26 letters of the ISO basic Latin alphabet, along with the Spanish  $\tilde{N}$  and the Ng digraph.

While vocabulary is centred around root words and the division between parts of speech is much more blurry than in Indo-European languages, it is still possible to distinguish nouns, verbs, adjectives, and adverbs, although typically only according to the applied affixes or the position in the sentence. Verbs are the most variable part of speech – they are subject to a system of over 80 affixes, and their form determines the semantic role ("focus") that the topic word plays in the sentence. There is no best choice for verbal lemma, because even in the infinitive there are still several possible lemmas per root word, each differing by the focus. If we listed only the root word in the dictionary (and redirect all the inflected forms to it), we would lose many important semantic distinctions, such as *bumili* ("to buy") and *magbili* ("to sell"), which would be conflated within a single entry for the root word *bili* (the broad concept of "exchange"), without the possibility of providing an explanation of the differences in meaning (and the respective translations). On the other hand, inflection in other parts of speech is very limited; instead, the language makes use of particles.

## 3. Dictionary structure

The structure of the dictionary is simple but comprehensive, each entry in the dictionary consists of:

- a headword,
- a list of inflected forms,

- a recorded pronunciation,
- a division into senses, with each sense comprising:
  - a disambiguating gloss,
  - where appropriate, one picture,
  - 1–10 collocations,
  - 1–10 synonyms, antonyms and related words,
  - three post-edited examples and up to 10 more (fully automatic),
  - English translations of the headword and one example and
  - Korean translations of the headword and one example.

# 4. Automatic dictionary drafting and post-editing

The automation procedure entirely relies on data, tools and methods we made available in Sketch Engine (Kilgarriff et al., 2014), a leading corpus management system. For Sketch Engine, we have crawled a 230-million-token corpus of Tagalog from the web and this has served as the basis for all the lexicographic work. While from the perspective of dictionary building the corpus is merely a needed by-product serving as empirical evidence for the automatic dictionary drafting, it represents a valuable linguistics resource as such (made available to general public through Sketch Engine), and to the best knowledge of the authors it is the biggest corpus for Tagalog as of July 2019.

The corpus was automatically part-of-speech tagged using Stanford PoS tagger (Toutanova et al., 2003)<sup>2</sup> and lemmatized using an in-house improved version of a Tagalog stemmer<sup>3</sup>. We also developed a sketch grammar so that related Sketch Engine's functions (mainly word sketches and thesaurus) become available.

For the post-editing phase of the 15,000 entries we used Lexonomy [Měchura, 2017], an open-source dictionary writing and editing tool. The editorial workflow consisted of isolated steps where editors were always post-editing only particular entry parts. In the following we explain in detail how individual entry parts were automatically generated and later post-edited. A dependency schema of the individual steps is provided in Figure 1.

<sup>&</sup>lt;sup>2</sup> Model was obtained from https://github.com/matthewgo/FilipinoStanfordPOSTagger

<sup>&</sup>lt;sup>3</sup> Available at https://github.com/crlwingen/TagalogStemmerPython



Figure 1: A dependency structure of all post-editing tasks.

	6	100
arts like this 🔻		
mayroo v		
malalim j		
hirap j		
sandali a		
desisyon n		
<b>madami</b> p		
non-word	W	<b>^</b>
foreign word	f	
wrong lemma	Ê	
wrong PoS	р	
Person name	п	
Organization name	0	
Other series dentity	е	
Other named entity		

Figure 2: Flagging inside of Lexonomy can be carried out with keyboard shortcuts our mouse clicks in the headword list.

### 4.1 Headwords

We have taken 45,000 most frequent corpus lemmas according to the document frequency. The editors have been validating them and removing non-words, foreign words, non-lemmas and proper nouns as well as correcting automatic part-of-speech tagging. The decision diagram for this task is given in Figure 7. The flagging feature was used for this task within Lexonomy (see Figure 2).

### 4.2 Inflected forms

Inflected forms were generated from the corpus based on the automatic lemmatization. Editors reassigned word forms to correct lemmas where necessary. Lexonomy features a built-in lay-by that behaves like an internal clipboard and is useful for moving entry parts across different entries.

### 4.3 Pronunciation

This is the only part of the entry that is done fully manually since there is no postediting of automatic text-to-speech output possible. On the other hand, it turned out to be also one of the simplest tasks.



Figure 3: Workflow of a word sense induction algorithm that exploits word-sketch-based collocations and word embeddings.

We provided the editors with a recording tool that they used in a small acoustic chamber. The tool prompted them to press a key to start a three-second recording window and then read a headword, after which the recording was automatically replayed to them and they had the option to revise it or move to the next headword.

In this scenario, the editors were able to record about 900 headwords per working day (8 hours). Afterwards, the recordings were automatically trimmed for silence and normalized using the Sox tool.<sup>4</sup>

# 4.4 Word sense division

Word sense clusters have been induced using a method that combines word sketches with word embeddings. The algorithm is to be presented in a separate paper in detail, but principally works as follows:

- for an input headword, take all its collocations, filtered by frequency (at least 5) and logDice score (Rychlý, 2008) (at least 2),
- for each collocation, take vectors of all words within a short window (4 words) across all collocation occurrences in the corpus and calculate the average of these vectors.
- cluster vectors obtained in previous step using HDBSCAN clustering (McInnes et al., 2017).

The workflow is also illustrated in Figure 3. The word embeddings were calculated on the source corpus using FastText (Bojanowski et al., 2016). The result of this procedure is a set of clusters, each consisting of one or more collocations, each being represented by a set of concordance lines in the sources corpus. Having each sense represented by a set of concordance lines is a very important principle that allowed us to proceed with many subsequent actions (e.g. example selection) on a per-sense level.

Editors were subsequently lumping and splitting the automatically induced clusters. Each cluster consisted of associated collocations and was backed by a set of concordance lines allowing users to inspect the underlying corpus evidence. For this task we have developed a custom editing widget for Lexonomy that is given in Figure 4. For every cluster, the editors may move the whole cluster or individual collocations into another sense or create a new sense. Alongside the senses, the editors were also post-editing English translations of the headword in each of its senses as well as assigning disambiguating glosses for each sense.

<sup>&</sup>lt;sup>4</sup> Available from http://sox.sourceforge.net/.

bago <sub>ADJECTIVE</sub> Senses:				Transl	ations:	
<ul> <li>sense 1 named: moderno</li> <li>sense 2 named: dati</li> </ul>		××	X	new	anono.	× 1 2
				modified		× 1 2
ADD SENSE				prior		× 1 2
				ADD T	RANSLATI	ION
				C	luster 1	
		Mark all:	1	2	NEW	V MIXED ERROR
example usage		а	ctions			collocate relation to headword concordance
mga bagong bayani	1	2	NEW	MIXED	ERROR	bayani <sub>NOUN</sub> nouns modified by "bago" of the second
ang bagong prinsesa	1	2	NEW	MIXED	ERROR	prrinsesa <sub>NOUN</sub> nouns modified by "bago"
bagong superhero	1	2	NEW	MIXED	ERROR	superhero_{NOUN} nouns modified by "bago" ${\mathscr O}$
				cl	uster 2	
		Mark all:	1	2	NEW	/ MIXED ERROR
example usage		actions				collocate relation to headword concordance
mga bagong sibol na	1	2	NEW	MIXED	ERROR	sibol <sub>NOUN</sub> nouns modified by "bago"
mga bagong halaman	1	2	NEW	MIXED	ERROR	halaman <sub>NOUN</sub> nouns modified by "bago" $\mathscr{S}$
bagong puno ng	1	2	NEW	MIXED	ERROR	puno <sub>NOUN</sub> nouns modified by "bago" &

Figure 4: A custom editing widget created for the purposes of post-editing word sense induction in Lexonomy.

#### 4.5 Disambiguating glosses

Disambiguating glosses (in Tagalog) were initially assigned when post-editing the word sense induction. Afterwards, they were reviewed by another editor and amended if necessary.

## 4.6 Pictures

Pictures have been automatically searched for in three online databases that offer API to access copyright-free images, namely Wikimedia Commons (Wikidata and Wiktionary) <sup>5</sup>, PixaBay <sup>6</sup> and Google Image Search <sup>7</sup> (only if no pictures were found in the previous two image sources).

The content of both Wikimedia Commons and PixaBay is copyright-free for the purposes of this project (being licenced as either CC0, CC-BY or CC-BY-SA). For Google Image Search we limited the search to pictures allowing commercial use with modifications.

 $<sup>^{\</sup>rm 5}$  See https://commons.wikimedia.org/wiki/Commons:API

<sup>&</sup>lt;sup>6</sup> See https://pixabay.com/service/about/api/

<sup>&</sup>lt;sup>7</sup> See https://www.googleapis.com/customsearch/v1

Initially, each sense was accompanied with ten images. Regrettably, English turned out to be the only reliable search language for all three engines we used. Afterwards, editors were selecting the best picture (up to three) out of the candidates, obtaining new images if necessary, as seen in Figure 5.



Figure 5: Post-editing interface for the selection of images matching the given word sense.

#### 4.7 Collocations

Collocations were initially obtained using word sketches in the word sense induction phase. However, because the clustering algorithm generally identified good representatives to separate clusters, there are typically many unclustered but still salient collocations. Therefore we have been adding, after the word senses were postedited, all high-scoring collocations if they were not clustered automatically. The goal was that for every grammatical relation in the word sketches, the top three collocations must be reviewed and added to the right sense if necessary.

It is important to emphasize the difference between using collocations as the vehicle for word sense induction (yielding clustered collocations) and making sure that all salient collocations are part of the entry and that this is not guaranteed by the word sense induction itself.

#### 4.8 Synonyms, antonyms and related words

Semantically related words were obtained using Sketch Engine's built-in thesaurus. We took advantage of having the word senses already post-edited and calculated the thesaurus on the sense level by adding another positional attribute indicating the sense

(based on the post-edited collocation occurrences). Early investigations have shown that the dominant sense prevails when looking up the thesaurus disregarding sense (i.e. just based on lemma and part of speech combination). On the other hand, such a sensedisregarding thesaurus tends to yield better results for the dominant sense (but not for other senses) because only a fraction of the collocations was typically clustered even for the dominant sense.

Therefore the editors were provided with the following data for each sense:

- top 10 items from a sense-disregarding (default) thesaurus
- top 10 items from a sense-based thesaurus.

Editors were then classifying all items into synonyms, antonyms, other related words (that are neither of the previous) and unrelated words in the post-editing phase.



Figure 6: Retrieving additional examples from within Lexonomy by calling Sketch Engine API.

## 4.9 Examples

Examples were generated using the GDEX functionality of Sketch Engine [Kilgarriff et al., 2008]. Editors selected the best of them or replaced them with new ones from the corpus using the pull model for interaction between Lexonomy and Sketch Engine (see Figure 6).

## 4.10 Translations

Translations to English were obtained automatically using Google Translate (which gives only one translation in the API) and Microsoft Bing (which can give multiple translations in the API), the results were merged and afterwards post-edited by translators. This happened during the post-editing of word sense for the translation of headwords/senses, and when post-editing the examples for the translation of the examples.

In the next phase, translations to English were validated by independent translators to assure their quality.

Translations from English to Korean were also carried out by post-editing machine translation output using the same commercial services. To be able to translate the isolated headwords/senses, the translators were carrying out that task together with translating the examples.

# 5. Editorial team and its post-editing workflow

Our team of editors consisted of seven adult native or near-native speakers of Tagalog, all with roots in the Philippines. They came from various social groups and had various occupations and educational backgrounds. All of them spoke both Tagalog and English, some also mastered another local language. In the recruitment process, we preferred the candidates *not* to be linguists, because the goal was to extract all the linguistic knowledge from the corpus and use human editors only to provide feedback on the quality of the machine-generated output and manually post-edit a selected portion of the entries. The Korean translations were commissioned to a professional Korean translator.

Work was distributed to the editors in batches in order to better account for individual needs. Before each new activity (such as headword annotation, proofreading of inflected forms, word sense division etc.), editors participated in a short training. For each activity, the content of the first batch was the same for everyone in order to check comprehension of the task, measure inter-annotator agreement and establish an average processing time per entry for each editor. The tasks were explained to the editors with as little linguistic terminology as possible, and the interface of the task-specific custom editors developed in Lexonomy was designed in order to reinforce this principle. For instance:

- In word sense division, the field to enter a disambiguating gloss was labelled in simple words: "sense name".
- In the list of collocations, the longest–commonest match representing a collocation was titled "example usage".
- Instead of being asked to regroup collocations among clusters (and actually feel that they are doing lumping and splitting), the editors were told to assign a sense number to each collocation in a list. This design choice has saved much clicking and the task could often be completed in a single pass.



Figure 7: A diagram showing the decision process when filtering an automatically produced headword list.

- The fact that the listed collocations were grouped in clusters was not commented on at all, as the only purpose thereof was to speed up the editors' work (putting collocations that presumably belong to the same sense next to each other) and no knowledge of the underlying logic was required on their part.

Inter-annotator agreement was measured after the completion of the first batch for each task. When that was done, we would usually summon the editors again and confront them about the patterns of disagreement in their output. At that moment, we would improve the written guidelines for the task (and possibly even reinvent the annotation process if necessary) which had only been sketched or non-existent until then. Following this meeting, editors would each be given a different set of data in order to speed up the process and cut on costs, but a small percentage of data would routinely be placed in two sets (either belonging to different editors, or subsequent sets belonging to the same editor) in order to monitor agreement and consistency throughout the whole process. Contact among editors was not discouraged – after all, they would spend time together during the training and some had already known each other before the start of the project – but attention had to be paid to prevent unwanted interdependence, particularly when all editors were working on the same set of data. On the other hand, we welcomed the creation of a chat group by the editors, which they could use for seeking and giving advice among themselves, both regarding the project's technical aspects and the linguistic uncertainties they had encountered during their independent work.

Only items (headwords, word senses, example sentences) that had been accepted in one postprocessing phase could advance into the following one. In spite of that, the editors would still occasionally discover wrong items at a later stage (such as being asked to review possible inflected forms of a word that is in fact not a lemma). This has served as an extra level of quality control and for each task, editors were instructed what to do when they come across such a case. As soon as the first headwords were completing their passage through the whole post-editing process and first entries emerging, we focused our attention back on the data that had been discarded or not yet available at the earlier stages: in close cooperation with the editors, we tried to fix errors in the lemmatization process manifested by the appearance of wrong lemmas in the list of headword suggestions. We would also consider any new headwords (or inflected forms) that may have emerged if we had increased the size of the source corpus since the start of the work. Any newly discovered headwords would then be submitted into the same pipeline as their predecessors, until there was no valid input left to be processed.

# 6. Conclusions

In this paper we report on a newly created Tagalog-English-Korean dictionary. The dictionary is fully corpus-based and the key novel aspect of its development is that the whole dictionary was initially created in a fully automatic way and afterwards manually post-edited where necessary. The post-editing phase presents many new challenges and

is far from being a finalized approach, but clearly shows its viability, affordability and performance benefits as for the time taken to produce the dictionary, which was about 9 months.

Overall the biggest challenge in this approach is to maintain solid data and user management rather than assuring sufficient quality of the automated outputs. The post-editing requires a lot of back-and-forth and trial-and-error, each being sensitive to careful data preparation and processing as well as being very communication intensive. More automation is clearly required to make these procedures robust, less error-prone and more affordable for less technically skilled lexicographers.

As for the automated tasks, it is worth mentioning that word sense induction turned out to be less of an issue than anticipated. The algorithm used tends to perform rather well for high-frequency polysemous words (but of course a more thorough evaluation should definitely be performed which was outside scope of our very practically motivated project). Throughout the tasks the importance of the size and quality of the corpus and its annotation was heavily manifested. We struggled a lot to crawl the at least 600-million word corpus, which we do not consider to be very big (although as far we know the biggest one for Tagalog). It was very obvious that a bigger corpus and better part-of-speech tagger and lemmatizer would improve the quality of the automated outputs as well as simplify some of the post-editing tasks a lot.

To summarize the issues we faced, data and user management were the major ones, then, less seriously, the corpus and its annotation, while all the automation procedures worked more or less as expected and did not cause any major issues.

Two more dictionaries are now in the pipeline following the same approach, where the source languages are Urdu and Lao. We continuously improve the tools and workflow, and will report on the other two dictionaries in a separate paper.

# 7. Acknowledgements

This work is part of a joint project with Naver Corporation. This work has been partly supported by the PhD Funding Scheme of Lexical Computing at Masaryk University. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731015. This work has been partly supported by the Grant Agency of CR within the project 18-23891S and the Ministry of Education of CR within the OP VVV project CZ.02.1.01/0.0/0.0/16 013/0001781 and LINDAT-Clarin infrastructure LM2015071.

## 8. References

- Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.
- Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlỳ, P.

& Suchomel, V. (2014). The sketch engine: ten years on. Lexicography, 1(1), pp. 7–36.

- Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. & Rychlý, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the* XIII EURALEX International Congress (Barcelona, 15-19 July 2008), pp. 425– 432.
- McInnes, L., Healy, J. & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), p. 205.
- Měchura, M. B. (2017). Introducing lexonomy: an open-source dictionary writing and publishing system. In I. Kosem et al. (eds.) Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference, 19-21 September 2017.
- Rychlý, P. (2008). A lexicographer-friendly association score. Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN, pp. 6–9.
- Toutanova, K., Klein, D., Manning, C. D. & Singer, Y. (2003). Feature-rich part-ofspeech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 173–180. Association for Computational Linguistics.

This work is licensed under the Creative Commons Attribution ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-sa/4.0/

