Semi-automatic building of large-scale digital dictionaries

[‡]Marek Blahuš, [‡]Michal Cukr, [†‡]Ondřej Herman,
[†‡]Miloš Jakubíček, [†‡]Vojtěch Kovář, [†‡]Marek Medveď

[†]Natural Language Processing Centre
Faculty of Informatics, Masaryk University

[‡]Lexical Computing
{firstname.lastname}@sketchengine.eu

Abstract

This paper presents a novel way of creating dictionaries by using a particular post-editing workflow, all of which is carried out in the context of building a set of three bilingual dictionaries – Tagalog, Urdu and Lao dictionaries with translations into English and Korean. The dictionaries were created completely from scratch without reusing any existing content and in a completely automatic manner, amounting to 50,000 headwords, out of which 15,000 headwords were subject to subsequent manual post-editing.

In the paper we discuss the post-editing methodology that we used and its impact on the overall lexicographic workflow. We describe the web corpora that were built specifically for the purpose of building these three dictionaries as well as their annotations (such as PoS tagging and lemmatisation) and tools that were used for the corpus annotation and for automating individual entry parts and the post-editing thereof. Most of the automatic drafting and post-editing relied on a backbone consisting of the Sketch Engine corpus management system and Lexonomy dictionary editor

We also detail the overall amount of work involved in each post-editing step, the technical and managerial difficulties faced alongside in the project, and the major technological issues that still need improvement in the post-editing scenario.

Keywords: post-editing lexicography; dictionary drafting; Sketch Engine

1. Introduction

Contemporary lexicography is based on using large text corpora to reflect the real use of a language as much as possible. Sometimes the corpora are only used as an additional tool helping lexicographers compile the entries, while other projects use corpora very extensively, generating large parts of the entries automatically and then post-editing, or correcting them. One of the most advanced procedures in the latter direction is the "Million-click dictionary" (MCD) method described in (Baisa et al., 2019) and (Jakubíček et al., 2020).

This paper reports on three related dictionary projects compiled using the MCD method, which are currently completed and signed-off. Looking back at the projects, we discuss the strengths and weaknesses of the approach, the errors made and lessons learned, and the overall resources needed to finish the projects.

2. About the Dictionaries

The three dictionaries are bilingual dictionaries from Tagalog, Urdu and Lao to English and further to Korean. Each dictionary consists of 15,000 manually post-edited entries and an additional 35,000 entries produced only automatically. Each post-edited entry contains:

- Pronunciation;
- possible word forms;

- sense disambiguators;
- translations into English and Korean;
- examples and their translation into English and Korean;
- an image (if appropriate);
- collocations;
- synonyms.

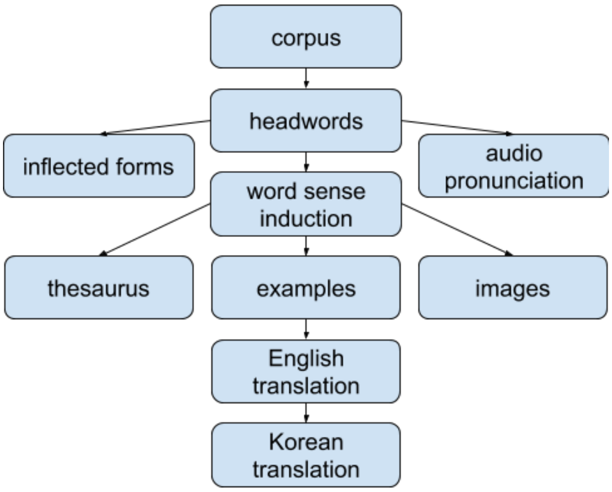


Figure 1: Workflow of the dictionary post-editing process

Following the MCD method, we divided entry creation into phases according to the entry parts above, and for each phase (except pronunciation) we generated data automatically from a large web corpus. Then the parts of the entries were manually cleaned and corrected by native speakers of the respective source languages; translations were proofread by translators. Each entry was in one phase at a time – the automatic data for the next phase were generated only after manual correction of the data in the previous phase. The overall workflow is demonstrated in Figure 1 and each of the steps is described in detail in the next section.

All post-editing steps have been implemented within the Lexonomy dictionary editor (Měchura, 2017) – typically as a custom editing widget, a small piece of JavaScript code a dictionary user can upload to set an editing form for an entry. This mechanism has proven to be sufficiently flexible and versatile to allow us to easily prepare a dedicated editing interface for a particular entry part.

3. Post-editing Workflow

3.1 Corpus processing

Three web corpora were created for the purposes of automatic dictionary drafting for each of the source languages using the methodology described in (Jakubíček et al., 2013). The sizes of the corpora were 230 (Tagalog), 265 (Urdu) and 120 (Lao) million words. Clearly, the sizes of the corpora are not overwhelming and represented a serious issue for automation, but we were simply unable to crawl more quality data from public websites. Our hypothesis is that for these languages most online content is published through non-open social networks instead of publicly available websites. Additionally, for all three languages, the internet contains a substantial amount of machine-translated content that has to be avoided as far as possible. For example, the Tagalog corpus contained 650 million words after initial boilerplate removal and partial deduplication; however, subsequent semi-automatic analysis of the data identified almost two-thirds of the data as machine-translated.

Each of the corpora was automatically part-of-speech tagged and lemmatised by different tools:

- for **Tagalog**, we used the Stanford tagger (Toutanova et al., 2003)[1] and lemmatised using an in-house improved version of a Tagalog stemmer[2],
- for **Urdu**, we used RFTagger (Schmid & Laws, 2008) to improve the tagging and lemmatisation output of the IIIT Hyderabad Urdu Parser[3],
- for **Lao**, we used RFTagger and a custom segmenter. Lao is not a flective language, thus lemmatisation was not relevant.

Additionally, we developed a word sketch grammar for each of the languages so that we could use the functions of the Sketch Engine (Kilgarriff et al., 2014) corpus management system.

3.2 Headwords

Headwords were automatically drafted by taking top words (lemmas) sorted by document frequency and having editors go over the list during post-editing. The classification manual used by the editors for Tagalog is provided in Figure 3. Editors labelled the headwords using the flag functionality in Lexonomy, as illustrated in Figure 2.

Additionally, the top 1,000 n-grams were also post-edited in order to cover the most salient multi-word expressions.

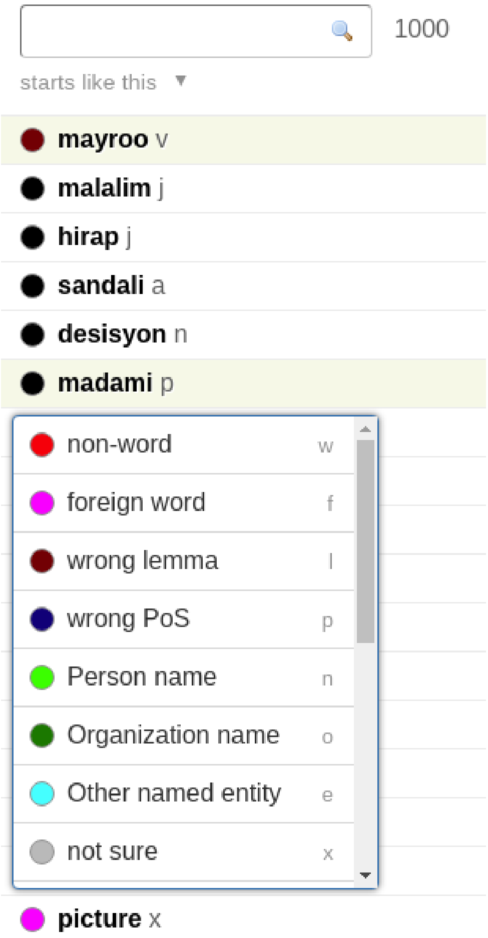
3.3 Inflected Forms

Inflected forms were automatically generated based on the automatic lemmatisation of the corpus. The editors reassigned inflected form where the lemmatiser incorrectly identified the base form.

3.4 Pronunciation

Pronunciation is the only part of the entry that was not automated. This is so for two reasons:

1. it would not be possible to "post-edit" the automatic recordings since there is no efficient way for a human to improve an automatically produced pronunciation in the form of an audio stream; and,
2. a manual recording can be carried out very quickly, so the potential gains of automation are rather limited.



● **mayroo** v

● **malalim** j

● **hirap** j

● **sandali** a

● **desisyon** n

● **madami** p

● **picture** x

Figure 2: Using flags for headword classification in Lexonomy

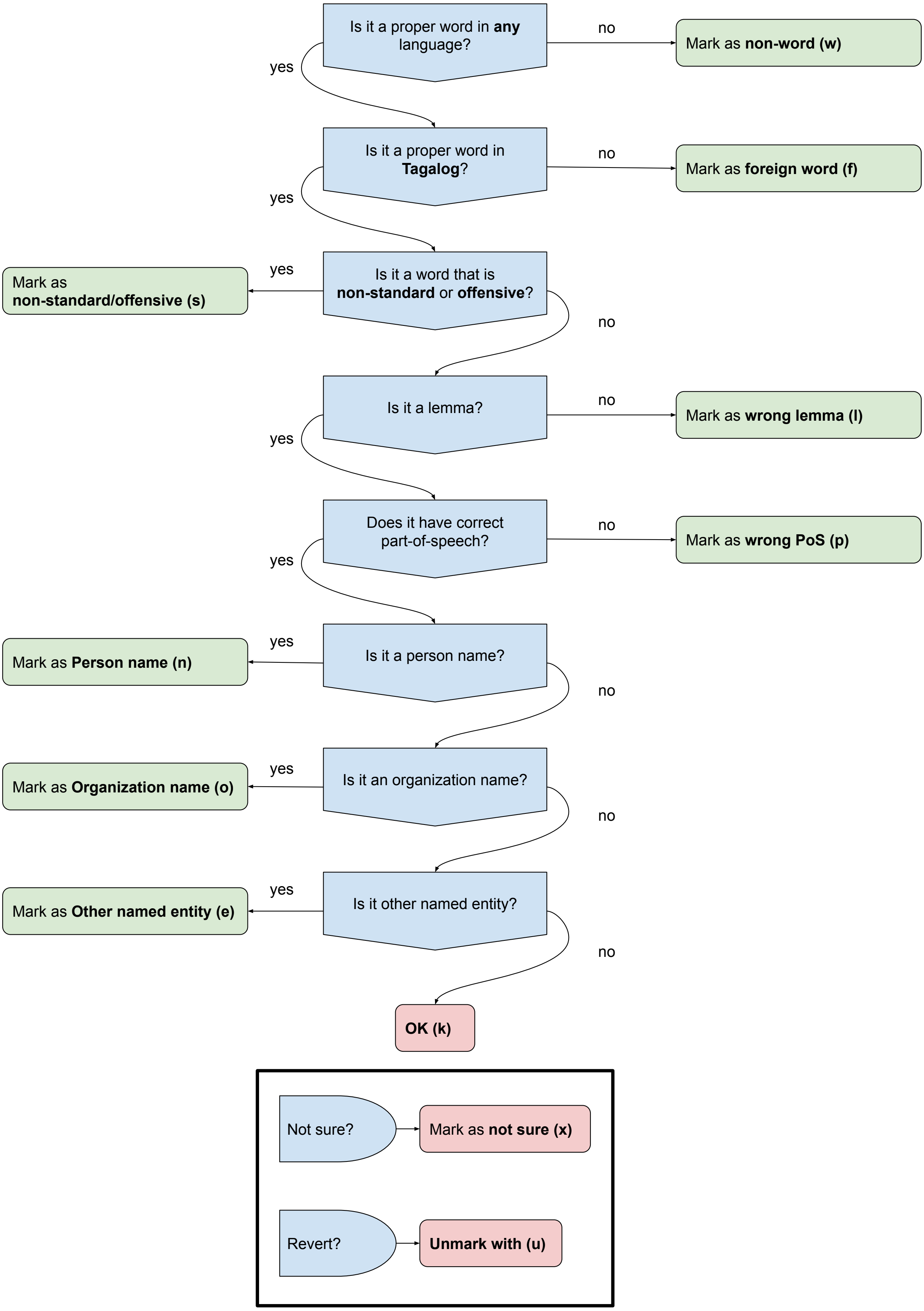

Is it a proper word in **any** language?
no
Mark as **non-word (w)**
yes
Is it a proper word in **Tagalog**?
no
Mark as **foreign word (f)**
yes
Is it a word that is **non-standard** or **offensive**?
yes
Mark as **non-standard/offensive (s)**
no
Is it a lemma?
no
Mark as **wrong lemma (l)**
yes
Does it have correct part-of-speech?
no
Mark as **wrong PoS (p)**
yes
Is it a person name?
yes
Mark as **Person name (n)**
no
Is it an organization name?
yes
Mark as **Organization name (o)**
no
Is it other named entity?
yes
Mark as **Other named entity (e)**
no
OK (k)
Not sure?
Mark as **not sure (x)**
Revert?
Unmark with (u)

Figure 3: Decision scheme for post-editing Tagalog headwords

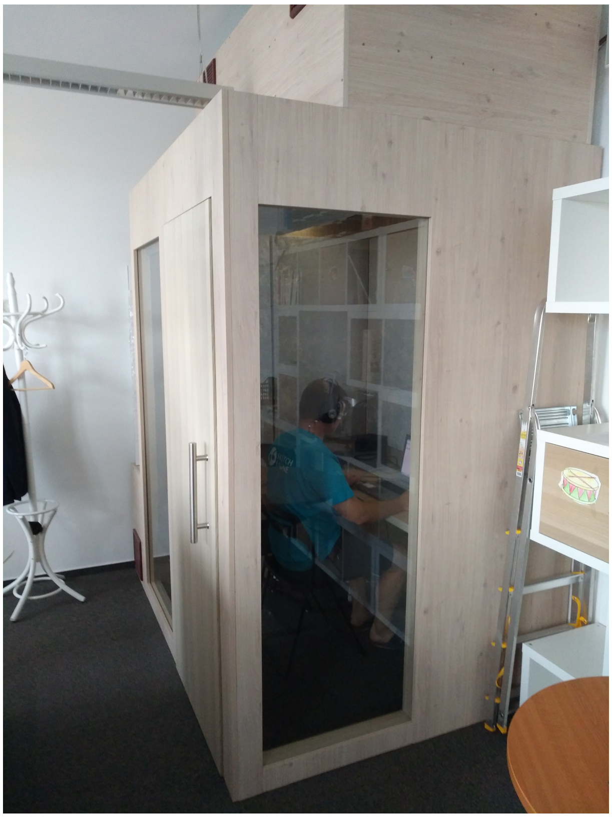


Figure 4: A sound-proof recording booth.

In our setting, native speakers recorded the pronunciation in a small recording booth (see Figure 4). They used a simple tool that displayed the next word to be recorded. A keyboard key press started the recording for a fixed amount of time (3 seconds); afterwards, the recorded sound was replayed to the speaker for confirmation (who then proceeded to the next word) or rejection (and the re-recording of the same word). This workflow allowed the speaker to record about 1,000 words during a working day of 8 hours, including inevitable rest breaks.

3.5 Word Sense Induction

We used a hybrid approach for identifying word senses by clustering the word sketch contexts according to the embedding vectors. We used skip-gram embeddings of dimension 300 using the fastText package (Joulin et al., 2016) and for every word sketch collocation of the examined word, we averaged collocation occurrence embeddings and used the HDBSCAN (McInnes et al., 2017) algorithm to cluster these vectors. The method is shown in Figure 5. HDBSCAN can determine the number of clusters automatically which is important because there is no reliable estimate for the number of word senses that we could use beforehand.

Editors were presented with the identified word sense clusters. Each cluster contained a set of collocations selected by the clustering algorithm. The editors reassigned

[1] The model was obtained from https://github.com/matthewgo/FilipinoStanfordPOSTagger
[2] Available at: https://github.com/crlwingen/TagalogStemmerPython
[3] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

collocations across the clusters/senses or created new senses. Along with of that process, disambiguators were created and translated into English. The editorial interface for this task can be seen in Figure 6.

After this post-editing step was finished, the corpus was sense-annotated by the senses/clusters as post-edited and this annotation was further used to generate a sense-based thesaurus, sense-based example sentences, and sense-based images.

3.6 Thesaurus

We obtained thesaurus items from the distributional thesaurus in Sketch Engine (Rychlý & Kilgarriff, 2007). The task for the editor during the post-editing phase was to validate each item for the particular word sense and classify them into synonyms and antonyms. Distinguishing between synonyms and antonyms is not yet automated and is a good candidate for further research.

3.7 Examples

Example sentences were obtained using GDEX (Kilgarriff et al., 2008) from Sketch Engine and validated and translated into English in the post-editing phase. This turned out to be one of the most tedious tasks in the end, owing to the very modest corpus sizes.

3.8 Images

We downloaded images from copyright-free online sources (Wikimedia projects, Pixabay, targeted Google Custom Search) and had the editors choose the best image (if any) for the particular word sense. The editing interface is demonstrated in Figure 7.

3.9 Translations

As the last step, disambiguators and example sentences were translated into Korean. Disambiguators were pre-translated using both Google Translate and Microsoft Bing; the latter was used mainly because it offers multiple translation candidates as part of its API. Unfortunately, it turned out that the alternative translation candidates given by Bing are just alternative word forms or spellings, so it did not help much to increase the diversity of the translation candidates before a human translator was validated them. Example sentences were translated using just Google Translate.

4. Data Management

We started the project with the idea of separated XML files, "batches" containing a few dozens of entries, which would fall through the annotation process as atomic units – and in the end we would just put them together into a dictionary. However, errors and disagreements in annotation (e.g. the example annotator refused to process the word previously accepted) led to a shrinking of the batches, complicated dependencies among them, the overall complexity of the data, and massive delays in processing.

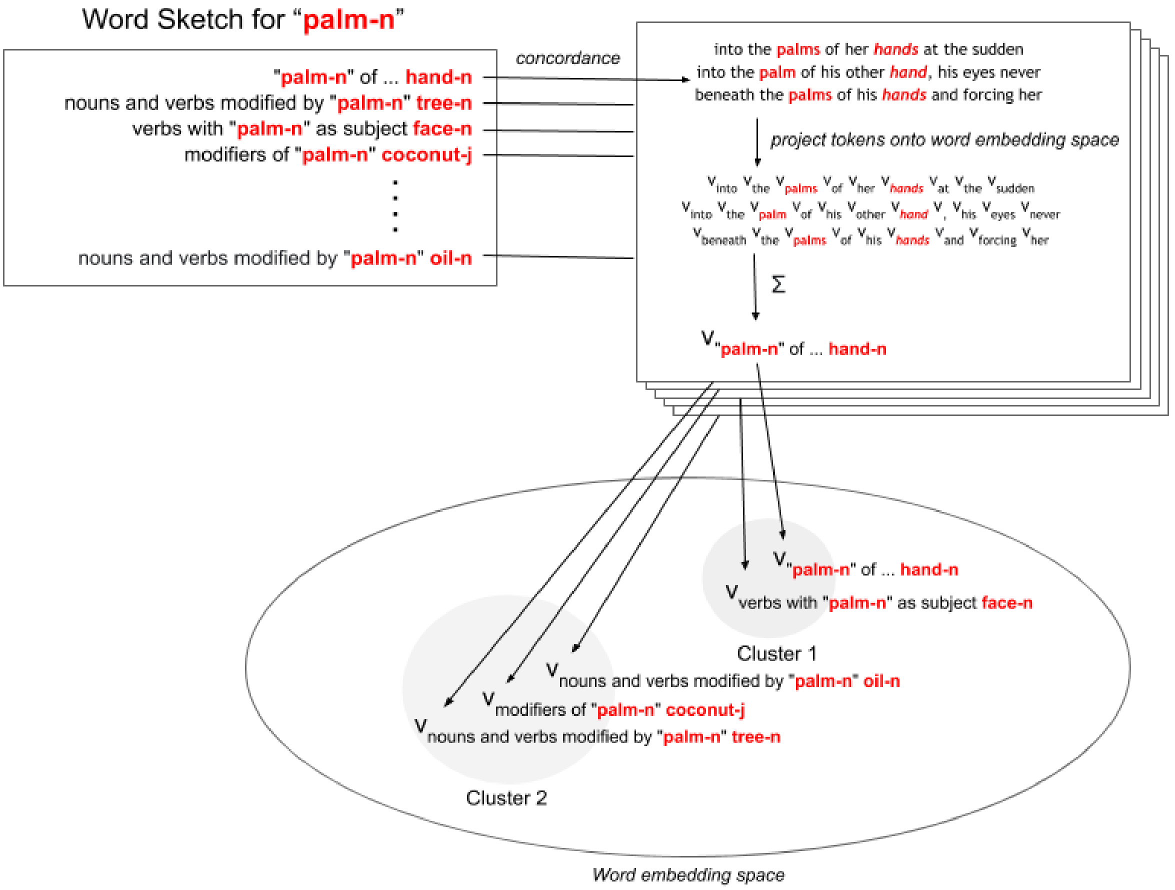

Word Sketch for "palm-n"
"palm-n" of ... hand-n
nouns and verbs modified by "palm-n" tree-n
verbs with "palm-n" as subject face-n
modifiers of "palm-n" coconut-j
nouns and verbs modified by "palm-n" oil-n
concordance
into the palms of her hands at the sudden
into the palm of his other hand, his eyes never
beneath the palms of his hands and forcing her
project tokens onto word embedding space
Σ
V"palm-n" of ... hand-n
Cluster 1
verbs with "palm-n" as subject face-n
nouns and verbs modified by "palm-n" oil-n
modifiers of "palm-n" coconut-j
nouns and verbs modified by "palm-n" tree-n
Cluster 2
Word embedding space

Figure 5: Using HDBSCAN over word sketch collocation embeddings

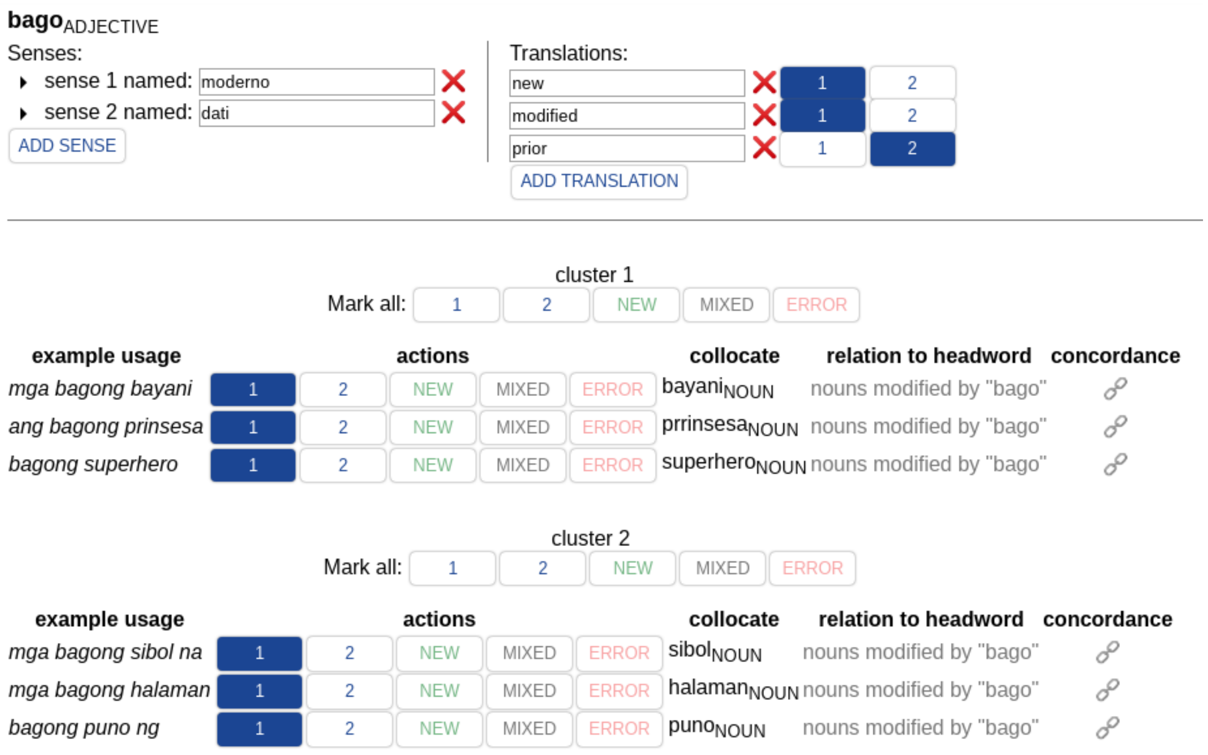
Therefore, we switched to a central database, stored in a novel textual format called NVH (name-value hierarchy)[4]. The batches for annotation were created as XML exports from this database, and finished annotations were processed as imports into the database. For each phase of entry processing, we implemented automatic import and export procedures that ensured consistency. The Git version control system was used so that it was subsequently possible to inspect, track, and fix problematic imports.

This mechanism worked much better and we managed to complete the dictionaries with it. However, there were still significant drawbacks:

- some of the annotation errors propagated and were only discovered when it was too hard to fix them (many of the fixes were done manually in the last phase of the project); and,
- there were errors in the original corpus annotation, so it was necessary to correct many of the headwords and propagate their correct form back to the corpus (so that the subsequent phases could be carried out correctly).

We find it crucial to understand that data management needs to be designed to take into account the inevitable human errors (and have a mechanism to handle them easily) and the fact that a source corpus is a noisy resource that the can be improved using

[4] http://www.namevaluehierarchy.org



bago$_{\text{ADJECTIVE}}$

Senses:

- ▸ sense 1 named: moderno ✗
- ▸ sense 2 named: dati ✗

ADD SENSE

Translations:

new	✗	[✓ 1]	2
modified	✗	[✓ 1]	2
prior	✗	1	[✓ 2]

ADD TRANSLATION

cluster 1

Mark all: 1 2 NEW MIXED ERROR

example usage	actions					collocate	relation to headword	concordance
mga bagong bayani	[✓ 1]	2	NEW	MIXED	ERROR	bayani$_{\text{NOUN}}$	nouns modified by "bago"	🔗
ang bagong prinsesa	[✓ 1]	2	NEW	MIXED	ERROR	prrinsesa$_{\text{NOUN}}$	nouns modified by "bago"	🔗
bagong superhero	[✓ 1]	2	NEW	MIXED	ERROR	superhero$_{\text{NOUN}}$	nouns modified by "bago"	🔗

cluster 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$_{\text{NOUN}}$	nouns modified by "bago"	🔗
mga bagong halaman	[✓ 1]	2	NEW	MIXED	ERROR	halaman$_{\text{NOUN}}$	nouns modified by "bago"	🔗
bagong puno ng	[✓ 1]	2	NEW	MIXED	ERROR	puno$_{\text{NOUN}}$	nouns modified by "bago"	🔗

Figure 6: Post-editing interface for word sense identification.

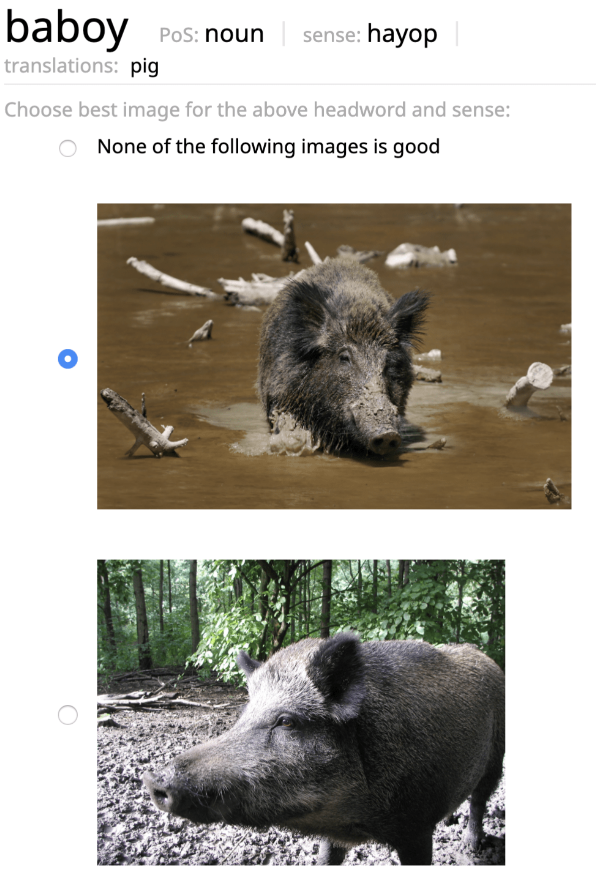
the annotations obtained in the post-editing phase. In our case, the automatic corpus annotation was improved whenever the annotators submitted corrections to part-of-speech tagging, lemmatisation or sense-identification. Updating the corpus and using its best version for further work was speeding up further post-editing tasks as well as created a better corpus, which for us was an important by-product in itself.

5. Time effort

The overall time effort for the Tagalog dictionary is available in Table 1. All three projects were started with approximately a 6-month lag and we managed to utilise the experiences gained as well as reuse many of the tools (such as the custom editing widgets in Lexonomy) so that the time effort for the Urdu dictionary (which started second) was about 20% less than for Tagalog, and for Lao (which started third) it was again about 20% less than for Urdu.

6. Conclusion

Before the start of the project execution, we mainly anticipated problems with the automatic algorithms generating the data – our main concerns were the possible low quality of the automatically generated data and therefore the low efficiency of the post-editing process. The reality was quite different. The output from these algorithms was mostly sufficient and the post-editing process was effective. We experienced the largest challenges in the management part of the project, and especially regarding the data management: keeping the data consistent, keeping the corpus consistent with the corrected data, keeping the annotation process running smoothly, and avoiding repeated cycles.



baboy PoS: noun | sense: hayop |
translations: pig

Choose best image for the above headword and sense:

○ None of the following images is good

●

○

Figure 7: Post-editing interface for images in Lexonomy.

Annotation phase	PH
Headwords	396
Revisions	464
Inflections	478
Audio (recording)	100
Senses + En translation	669
Collocations	204
Images	313
Thesaurus	617
Examples + En translation	1,938
Examples proofreading	135
Examples corrections	373
Translation into Korean	772
Final review	591
Final manual changes	87
Training, communication	64
Total	7,199

Table 1: Person-hours spent on annotation for the different phases of the Tagalog dictionary

In further projects, this is the part that needs to be focused on in the first place: solving this successfully is key to the overall success of the project.

The projects have also reconfirmed the importance of corpus size for quality lexicographic work. The sizes of the corpora used should be seen as the necessary minimum and many of the issues we faced would not be present if the corpora had been, for example, 10 times bigger, which would easily be the case for many better-resourced languages.

Overall, the projects clearly showed the vitality of the post-editing workflow in lexicography as well as the technological readiness of the lexicographic tools that we used. We are confident that further improvements in the management of the whole process can bring further significant savings as regards the in time effort required.

Acknowledgements

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 Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

7. References

Baisa, V., Blahuš, M., Cukr, M., Herman, O., Jakubíček, M., Kovář, V., Medveď, M., Měchura, M., Rychlý, P., Suchomel, V. (2019). Automating dictionary production: a Tagalog-English-Korean dictionary from scratch. In *Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference. 1-3 October 2019, Sintra, Portugal.* Lexical Computing, pp. 805–818.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P. & Suchomel, V. (2013). The TenTen Corpus Family. *International Conference on Corpus Linguistics, Lancaster.*

Jakubíček, M., Kovář, V. & Rychlý, P. (2020). Million-Click Dictionary. In *Proceedings of XIX EURALEX Congress: Lexicography for Inclusion, Vol. II [to be published].*

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. *CoRR*, abs/1607.01759. URL http://arxiv.org/abs/1607.01759. 1607.01759.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography*, 1. URL http://dx.doi.org/10.1007/s40607-014-0009-9.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M. & Rychlỳ, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX international congress.* Documenta Universitaria Barcelona, Spain, pp. 425–432.

McInnes, L., Healy, J. & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11).

Měchura, M.B. (2017). Introducing Lexonomy: an open-source dictionary writing and publishing system. In *Electronic Lexicography in the 21st Century: Lexicography from Scratch. Proceedings of the eLex 2017 conference, 19-21 September 2017.*

Rychlý, P. & Kilgarriff, A. (2007). An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 41–44.

Schmid, H. & Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 777–784.

Toutanova, K., Klein, D., Manning, C.D. & Singer, Y. (2003). Feature-rich part-of-speech 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*. Association for Computational Linguistics, pp. 173–180.