Dr. Livingstone, I presume? Polishing of foreign character identification in literary texts

Aleksandra Konovalova
University of Turku
aleksandra.a.konovalova@utu.fi

Kristiina Taivalkoski-Shilov
University of Turku
kristiina.taivalkoski-shilov@utu.fi

Antonio Toral
University of Groningen
a.toral.ruiz@rug.nl

Abstract

Character identification is a key element for many narrative-related tasks. To implement it, the baseform of the name of the character (or lemma) needs to be identified, so different appearances of the same character in the narrative could be aligned. In this paper we tackle this problem in translated texts (English–Finnish translation direction), where the challenge regarding lemmatizing foreign names in an agglutinative language appears. To solve this problem, we present and compare several methods. The results show that the method based on a search for the shortest version of the name proves to be the easiest, best performing (83.4% F_1), and most resource-independent.

1 Introduction

Character identification is both a complex and a difficult task that can be solved using different methods, from manual (Declerck et al., 2012) to automatic (Goh et al., 2012). One of the necessary steps for character identification is to detect which exact character appears in the text (Labatut and Bost, 2019). For such detection, lemmatization is required.

Lemmatization is a process of assigning to a wordform its lemma (Kanerva et al., 2019). It is one of the important tasks in Natural Language Processing (henceforth NLP), since many other NLP methods require it during the preprocessing stage. For agglutinative languages, such as Finnish, correct lemmatization can turn out to be a difficult task because one word can have many wordforms (e.g. a Finnish word may have more than 50 wordforms. Consider an example for English name *Lizzy* in Finnish translation: *Lizzy*, *Lizzystä* (from / about Lizzy), *Lizzylle* (to Lizzy), *Lizzyn* (Lizzy's)). Current state-of-the-art models that use Neural Networks can help with solving this task. For example, such a lemmatization model is implemented as part of the Turku neural parser pipeline, which currently yields the best results for Finnish lemmatization (Kanerva et al., 2019). However, their accuracy, though close to 100%[1], is not perfect, so lemmatization may require further refinement which would help to enhance the end result for character identification.

In this paper we discuss enhancing foreign characters' identification for English characters in Finnish texts, via improving lemmatization of characters' names. The structure of the paper is as follows: first we provide an overview of the related work (Section 2), subsequently we describe our data (Section 3), after which we discuss the creation of the gold standard for our methods and a definition of character in the context of our research (Section 4). We continue the paper with describing the methods (Section 5) that we introduced and used. Finally, we present our results and analyze them (Section 6). We conclude our paper in Section 7. Code for the paper is available at https://github.com/AleksanKo/naacl2022.

2 Related work

Lemmatization for agglutinative languages, such as the one targeted in our study (Finnish), has been tackled from different perspectives. The first attempts to solve the problem for Finnish were the FINTWOL tool (Koskenniemi, 1983) and Morfo (Jäppinen and Ylilammi, 1986). Around three decades later one of the most known non-neural methods, Omorfi (Pirinen, 2015) was developed. Omorfi uses finite state transducers and can be used further for enhancing lemmatization (Silfverberg et al., 2016). The current state-of-the-art is represented by Turku neural parser pipeline (Kanerva et al., 2018), (Kanerva et al., 2019) that treats lemmatization as a sequence-to-sequence problem using the OpenNMT neural machine translation

[1]It ranges from 95.1% to 97.7%, depending on Finnish morphological annotation it is applied to (Kanerva et al., 2019).

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pages 123 - 128
July 10-15, 2022

toolkit (Klein et al., 2017) and yields 95%-97% accuracy.

In our research, we are focusing on retrieving canonical forms of foreign names. This may be challenging since foreign names are not typically expected by the lemmatizer, so it may be prone to errors. However, this step is necessary in case of agglutinative languages: otherwise one character may split into two or more characters (for example, instead of *Catherine*, we would have three entities: *Catherine*, *Catherinea* and *Catherinen*), which affects further the results for building character networks or narrative.

3 Data

The data used in our experiments is a corpus of Finnish translations made by Kersti Juva (a subcorpus of the *Classics of English and American Literature in Finnish* corpus, or *CEAL*[2]). The corpus consists of the short novel *Washingtonin Aukio*, 2003 ("Washington Square", 1880) by Henry James and the novels *Ylpeys ja ennakkoluulo*, 2013 ("Pride and Prejudice", 1813) by Jane Austen and *Kolea talo*, 2006 ("Bleak House", 1853) by Charles Dickens. The corpus is stored as text files, 3 files and 384,053 words in total.

4 Creation of gold standard

Before applying our methods (see Section 5), we had to choose a gold standard character names' list, so that we can evaluate our methods. To perform this task, we got the information from different internet sources that contain information about characters from the novels in our dataset (see Appendix A).

While creating a gold standard character names' list, we also faced many questions about characters, such as: what is a literary character? Who do we consider a character from the point of the narrative? Who do we consider a character from the point of character extraction where we are forced to filter the results of automatic Named Entity Recognition[3]? Do we take into consideration off-screen characters (characters that are only mentioned in the text and do not participate in the plot)? To answer these questions, we need to define what / who the character is.

[2]https://www.kielipankki.fi/corpora/ceal-2/

[3]This is part of the preprocessing used in our experiments, see Section 5.

The literary character can be seen as a construct whose definition and features depend on the study area (Margolin, 1990). Jannidis (2013) considered a character "a text- or media-based figure in a storyworld, usually human or human-like" or "an entity in a storyworld". Overall, characters are intertwined with narrative and storyworld, contributing to their development from many aspects.

We considered a literary character every figure that was relevant for the narrative development (thus, e.g. names of famous persons that are mentioned but do not appear in the novel were not included). So we decided to include both onscreen (entities that are actively participating in the storyworld) and off-screen (entities that are passively contributing to the construction of the storyworld) characters (e.g. in case of *Washington Square*, it was the mother of the main character that gets mentioned only twice). We also included all possible names that can be used for naming a certain character by splitting the full name (e.g. *Elizabeth Bennet* would also get versions *Elizabeth* and *Bennet*) and by analyzing possible versions (*Lizzy* for *Elizabeth Bennet*) that were mentioned in the internet sources (see Appendix A). So *Elizabeth Bennet* would get the following names: *Bennet*, *Eliza*, *Eliza Bennet*, *Elizabeth*, *Elizabeth Bennet*, *Lizzy*. The creating of the gold standard was carried out only by one annotator.

5 Methods

To apply our methods, we have to carry out the preprocessing first. This includes the following workflow:

1. Applying Named Entity Recognition on Finnish translations of English texts;
2. Filtering the named entities identified by label, removing all entities that are not persons;
3. Getting the lemmas for the remaining named entities.

For Named Entity Recognition on Finnish texts and further lemmatization, a Python library named Stanza (Qi et al., 2020) was used, because it provided a state-of-the-art level of Named Entity Recognition for this language. Finnish language models are represented in Stanza by Turku neural parser pipeline (Kanerva et al., 2019), so we will be using the Turku neural parser pipeline's lemmatizer (Kanerva et al., 2018) as a baseline.

We have used and compared three methods of finding correct names' lemmas. These methods were applied on the output of the preprocessing, i.e. lists of names that were results of applying Named Entity Recognition on Finnish translations, then filtering only person-type entities, and finally lemmatizing them. The methods were implemented using Python and are as follows:

1. *Method 1*: Check for the shortest version of the name. This method was based on two assumptions: 1) that the language is agglutinative, so the stem is modified a lot with the help of affixes, and 2) that a character name will appear many times, so not all its wordforms contain morphemes from the target language and there will be at least one occurrence of the correct lemma. Consider the following example: if we have the right lemma of the character name (*Catherine*) and wrong versions that were however recognized as lemmas by the lemmatizer (*Catherinen*, *Catherinea*), the right version is the shortest, so searching in the sorted list of names [*Catherine*, *Catherinea*, *Catherinen*] should yield the right result.

2. *Method 2.1* and *Method 2.2*: Check whether the name exists using Wikipedia[4] or Wiktionary,[5] respectively (in our case, the English version of these resources). This method requires that for most of the names there were articles in Wikipedia and in Wiktionary, and since we were using English versions of these resources, wrong forms that contained Finnish suffixes would be discarded. This assumption relied heavily on the genre of texts of the corpus, namely classic British and American literature, so the character's name was an actual name in the real world. If we consider the example from *Method 1*, *Catherine* would return an article from both Wikipedia and Wiktionary, while *Catherinen* and *Catherinea* would return an error which means that there was no such page, and, presumably, no such name in the English language.

3. *Method 3*: Check if the word occurrence contains suffixes (in our case, Finnish suffixes). In this implementation only suffixes corresponding to Finnish genitive and partitive cases were checked, since the lemmatizer usually made mistakes in such forms. For example, if we check for the words that end on *-a/ä* and *-n*, the wrongly lemmatized *Catherinen* and *Catherinea* would not be included in the end results.

[4]https://en.wikipedia.org/
[5]https://en.wiktionary.org/

6 Results

We evaluated the results of our methods and the baseline according to the following criteria:

- Precision (fraction of true positive instances among all extracted instances), recall (fraction of true positive instances that were retrieved) and F-score (harmonic mean of precision and recall).

- Language independence (whether the method depends on certain language features and / or language resources, such as corpora, or not).

- Need for external non-linguistic resources (whether the method requires external resources to perform checking or not).

The overall count of results can be found in Table 1. The *Gold standard* column contains the number of character names (number of true positive instances, or all possible versions that can be used for naming all the characters that appear in the novel), *Method 1* covers results for checking for the shortest version of the name, *Method 2.1 and Method 2.2* - for checking in Wikipedia / Wiktionary, and *Method 3* - for checking for suffixes.

In Table 2 and Table 3 we present the results for the precision and recall for each method and the baseline, respectively. The results for F-score were counted only on average level and can be seen in Table 4.

It is quite noticeable from Table 1 that Method 2.2. (search for a correct wordform using Wiktionary) usually retrieves less names than any of the other methods (it has the lowest count of names for *Bleak House*, and the second lowest for *Washington Square*). However, in terms of recall, which can be seen in Table 3, the results varied significantly for this method: from 46% to 92% (compared with other methods where recall did not go lower than 55%).

Method 1 performed well on both a short text (*Washington Square*) and a significantly longer novel (*Bleak House)*. It reached 100% recall for

Work	Gold standard	Baseline	Method 1	Method 2.1	Method 2.2	Method 3
Washington Square	20	22	18	21	17	12
Pride and Prejudice	48	80	76	65	60	65
Bleak House	126	184	128	88	68	109

Table 1: Names count for the three methods and the baseline

Work	Baseline	Method 1	Method 2.1	Method 2.2.	Method 3
Washington Square	73%	89%	76%	88%	**92%**
Pride and Prejudice	59%	63%	69%	**73%**	65%
Bleak House	58%	84%	**85%**	**85%**	83%
Average precision	63.3%	78.7%	76.7%	**82.0%**	80.0%

Table 2: Precision of the three methods comparing to the baseline. Average precision is added for reference. Best result in each row shown in bold.

Work	Baseline	Method 1	Method 2.1	Method 2.2.	Method 3
Washington Square	**80%**	**80%**	80%	75%	55%
Pride and Prejudice	98%	**100%**	94%	92%	88%
Bleak House	85%	**86%**	60%	46%	72%
Average recall	87.7%	**88.7%**	78.0%	71.0%	71.7%

Table 3: Recall of the three methods comparing to the baseline. Average recall is added for reference. Best result in each row shown in bold.

Pride and Prejudice, but precision for this text was lower than for other two: 63%.

Both external sources that were used for Method 2 (Wikipedia and Wiktionary) showed the worst recall results on *Bleak House* (46% and 60%) but scored over 90% on *Pride and Prejudice*. In terms of precision, checking in Wiktionary (Method 2.2) performed better than using Wikipedia for both *Washington Square* and *Pride and Prejudice*, while the usage of both resources led to the same result for *Bleak House*. We assume that this result can be attributed to the difference between the names, surnames and nicknames used in these novels.

Method 3 achieved the second best precision overall (and the best precision for *Washington Square*), but did not show good results in terms of recall (worst for two texts out of three). While applying this method, we also noticed that, without applying additional checks, it seems to filter out a certain amount of true positive cases, since the suffixes in question (partitive and genitive) contain one or two letters and can easily be just parts of correct lemmas.

In Table 4 we present values for the aforementioned criteria of evaluation, i.e. language independence and need for other resources, as well as the average precision, recall and F-score.

Only one method can be considered language-independent: search for the shortest version of lemma (Method 1). It can also be considered the only method that does not require a lot of external sources of knowledge, since even searching for the suffixes requires knowledge of Finnish grammar. The only knowledge that is required for the first method is knowledge about the type of language (agglutinative / fusional), but since the problem with wrongly lemmatized names is mostly the problem of agglutinative languages, this knowledge can be considered basic.

It is worth noting that lemmatization and scrupulous study of extracted names has also shown changes in translation regarding the original text. Thus, there is no *Guster* (the servant of Mr. Snagsby and Mrs. Snagsby) in the Finnish version of Bleak House but *Molly*, due to the wordplay. Such changes made the creation of the gold standard more difficult since it was based on the original namings of characters. We suggest that

Criteria	Baseline	Method 1	Method 2.1 / Method 2.2	Method 3
Average precision	63.3%	78.7%	76.7% / **82.0%**	80.0%
Average recall	87.7%	**88.7%**	78.0% / 71.0%	71.7%
Average F-score	73%	**83.4%**	77.3% / 76.1%	75.6%
Language independence	-	partly yes	no	no
External resources	-	no	Yes, database	Yes, linguistic knowledge

Table 4: Comparison of the methods (also regarding the baseline). Best result in each row shown in bold.

word alignment with original texts could help find such cases automatically. However, word alignment would not solve the lemmatization in these cases, since the name in the original (English) and in the translation (Finnish) differ.

There were also some issues related to misprints in the Finnish translations (e.g. in the translation of Washington Square sometimes names *Lavinia* and *Catherine* were misprinted as *Lavina* and *Catherna*) which lead to additional wrong results. Such errors were fixed, so the final version of results contained only right versions of names.

7 Conclusion

Perhaps surprisingly, a rather simple method that searches for the shortest version of the character's name (Method 1) yielded one of the best results with average precision of 78.7%, the best overall recall (88.7%) as well as the best overall F_1 (83.4%).

Searching for a name in Wikipedia (Method 2.1) led to slightly lower precision (77.6%). Searching for a name in Wiktionary (Method 2.2) was overall slightly worse than Method 2.1 (F_1 76.1% vs 77.3%), but almost on the same level as checking if the name contains suffixes (Method 3): average precision for both was about 71%.

In addition, Method 1 did not require any additional resources and it was relatively language-independent which would allow it to be used without any modifications for other agglutinative languages. We suggest that a combination of these methods (for example, simple combination of Method 1 and Method 3 should help e.g. in case when the characters do not have common names in genres like fantasy or sci-fi) will further improve the search for the right lemmas for foreign names in texts written in agglutinative languages and thus enhance the character identification.

Acknowledgements

This research was supported by the TOP-Säätiö (*TOP Foundation*).

References

Thierry Declerck, Nikolina Koleva, and Hans-Ulrich Krieger. 2012. Ontology-based incremental annotation of characters in folktales. In *LaTeCH@EACL*.

Hui Ngo Goh, Lay Ki Soon, and Su Cheng Haw. 2012. Automatic identification of protagonist in fairy tales using verb. In *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Proceedings*, number PART 2 in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 395–406.

Fotis Jannidis. 2013. Character. In Peter Hühn et al., editor, *the living handbook of narratology*. Hamburg University, Hamburg.

Harri Jäppinen and Matti Ylilammi. 1986. Associative model of morphological analysis: An empirical inquiry. *Comput. Linguist.*, 12(4):257–272.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 133–142, Brussels, Belgium. Association for Computational Linguistics.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2019. Universal lemmatizer: A sequence to sequence model for lemmatizing universal dependencies treebanks. *CoRR*, abs/1902.00972.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Kimmo Koskenniemi. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Number 11 in Publications.

University of Helsinki. Department of General Linguistics, Finland.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *CoRR*, abs/1907.02704.

Uri Margolin. 1990. The what, the when, and the how of being a character in literary narrative. *Style*, 24:453–68.

Tommi A Pirinen. 2015. Development and use of computational morphology of finnish in the open source and open science era: Notes on experiences with omorfi development. *SKY Journal of Linguistics*, 28:381–393.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *CoRR*, abs/2003.07082.

Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén, and Mikko Kurimo. 2016. Finnpos: an open-source morphological tagging and lemmatization toolkit for finnish. *LANGUAGE RESOURCES AND EVALUATION*, 50(4):863–878.

A Sources

1. The 5 Least Important Characters in Pride and Prejudice, accessed 09.01.2022. https://theseaofbooks.com/2016/04/29/the-5-least-important-characters-in-pride-and-prejudice/

2. Austenopedia, accessed 09.01.2022. http://austenopedia.blogspot.com/p/entry-number-1.html

3. Bleak House Characters | Course Hero, accessed 09.01.2022. https://www.coursehero.com/lit/Bleak-House/characters/

4. Washington Square Character Analysis | LitCharts, accessed 09.01.2022.https://www.litcharts.com/lit/washington-square/characters