Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Integrating Semantic Embeddings into NoRaRe

Arne Rubehn
Chair for Multilingual Computational Linguistics
University of Passau

This study illustrates how semantic embeddings can be added to and retrieved from NoRaRe. By that, it provides a template for handling vector data and makes popular methodology in semantic modeling available for cross-linguistic comparison.

1 Introduction

The success of word embedding techniques has lead to a complete shift to a distributional framework in computational semantics, which is underlined by the fact that word embeddings build the backbone of modern Large Language Models. Following the distributional hypothesis, vector representations for words (i.e., embeddings) are learned from surrounding words. Words with similar meanings and/or functions will therefore appear in similar contexts, leading to similar representations, while words that are found in different contexts are considered dissimilar and thus appear distant from each other in the embedded vector space. Since they are demonstrably powerful, yet readily interpretable, static word embeddings (meaning that one word maps to one vector representation, in contrast to contextual word embeddings) trained with models like Word2Vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014), and FastText (Bojanowski et al. 2017) remain a popular choice for the computational investigation of semantic similarity and relations between word forms.

While certainly powerful and highly expressive, word embeddings cannot be directly plugged into models for computer-assisted language comparison. Research in comparative linguistics relies on the presence of comparative concepts — items under the same gloss in different languages should refer to the same sense or concept (even if the gloss in the metalanguage is potentially ambiguous). For example, the English word *bark* can refer to the exterior part of a tree or the vocalization of a dog. The word embedding for *bark* captures both meanings — simply using that embedding for a

Computer-Assisted Language Comparison in Practice
Volume 8, Number 2
URL: https://calc.hypotheses.org/8783 DOI: 10.15475/calcip.2025.2.3

Published on 17/09/2025

wordlist where *bark* only refers to the part of the tree would thus lead to an imprecise representation of the expressed concept due to the homonymy in the English gloss. To address this problem, we recently presented a new technique for training language-agnostic *concept embeddings* from cross-lingual colexification networks (Rubehn and List 2025).

In this post, I will describe how those concept embeddings were integrated into the Database of Norms, Ratings, and Relations (NoRaRe, Tjuka et al. 2022, https://norare.clld.org/) alongside multilingual FastText word embeddings (Grave et al. 2018). By that, I will provide a general tutorial on how to integrate and retrieve vector data with NoRaRe.

2 Integrating Concept Embeddings into NoRaRe

Following last month's tutorial by List (2025), we start by expanding the `norare.tsv` and `datasets.tsv` files that contain basic information about all datasets collected in NoRaRe, as well as adding new references to `references/references.bib`. In `datasets.tsv`, I simply append one new line where I provide meta-information concerning the new dataset under the name `Rubehn-2025-ConceptEmbeddings`. This name is used as unique identifier for the dataset throughout the entire integration process. Similarly, I extend `norare.tsv` by describing the type of data I am contributing. Since for each concept I am contributing several vector representations (3 types of concept embeddings based on different types of colexification + FastText embeddings in 9 different languages), each of them has to be described separately in a new line. Finally, I add BibTeX entries for the sources of the data I am contributing (in this case Grave et al. 2018; Rubehn and List 2025) to `references/references.bib`.

Having dealt with the basic files, it is time to contribute the actual dataset. For this, the first step is to create a new directory `datasets/Rubehn-2025-ConceptEmbeddings` where all relevant information is stored. This directory itself will contain three files: `Rubehn-2025-ConceptEmbeddings.tsv`, `Rubehn-2025-ConceptEmbeddings.tsv-metadata.json`, and `norare.py`. Note how the dataset identifier is recurring in the namespace.

`Rubehn-2025-ConceptEmbeddings.tsv-metadata.json` is a metadata file conforming to CSVW standards (Gower 2021). Simply speaking, it allows (and requires) me to define which columns the corresponding TSV file (where all data is written to) contain, and which data type is expected in each column. This ensures data consistency and allows for smooth data retrieval later on. Concretely, I use this file to define that each entry consists of a Concepticon ID, the corresponding Concepticon gloss (the concept inventory that NoRaRe operates on is defined by Concepticon; List et al.

2025, https://concepticon.clld.org/), and then all different types of embedding vectors described above. For all columns containing embeddings, I define `"json"` as data type: embeddings are therefore directly represented as lists of floats.

Finally, we get to the core of the dataset: the `norare.py` script that produces the file `Rubehn-2025-ConceptEmbeddings.tsv`, which contains all the actual data in the end. In `norare.py`, I have to define two functions that correspond to shell commands defined by PyNorare (List and Forkel 2024): `download` and `map`. As the names suggest, these two functions define the behavior for 1. downloading the raw data and 2. mapping them to NoRaRe.

For the concept embeddings, it is straightforward to define the downloading and mapping behavior, since the embeddings already represent Concepticon concepts directly. The only trick I had to employ here was to round all numbers to 4 decimals to prevent the file from becoming too large. This is a simple quantization technique that is commonly done to compress data (Gray and Neuhoff 1998) without losing relevant information.

3 Integrating Multilingual Word Embeddings into NoRaRe

Mapping multilingual FastText embeddings (Grave et al. 2018) requires some more handling of data, which is essentially due to the fact that embedded words have to be mapped to the concepts defined by Concepticon, and this mapping is usually not a 1-to-1 relation. As a first step, again, we download the data from https://fasttext.cc/docs/en/crawl-vectors.html.

Now that we have obtained the embeddings for words in different languages, we have to map them to the relevant concepts. This can be done via the MultiSimLex database (Vulić et al. 2020) that provides parallel translations for 1,888 cues in various different languages. Since the elicited cues have already been linked to Concepticon (List 2021) and the translations are parallel, MultiSimLex offers a good resource for systematically linking words from different languages to Concepticon concepts.

Now that we have a principal method for mapping words to concepts, we encounter another problem that we need to deal with: sometimes, there are multiple translations for the same concept. For example, the concept CAR can be expressed by the Russian words *avtomobil'* and *mashina*, and both translations are actually found in the MultiSimLex data. In those cases, we define the vector representing a concept in a language as the weighted average of the corresponding word vectors. As a simplified example, assume that we find 1 occurrence of *avtomobil'* with the vector `[2, 3]` and two occurrences of *mashina* with the vector `[1, 6]`: the resulting vector would be `[(1*2 + 2*1) / 3, (1*3 + 2*6) / 3] = [1.3333, 5]`.

With solutions to these practical issues, we now have a robust and consistent way of handling and mapping word embedding data to Concepticon; so we can simply implement the described behavior in the map function.

Having implemented both the downloading and mapping routine for both, the concept embeddings and the word embeddings, pynorare offers convenient shell commands to create the desired TSV file, in which all information is finally represented.

```
$ norare download Rubehn-2025-ConceptEmbeddings  # download data
$ norare map Rubehn-2025-ConceptEmbeddings       # map data
```

4 Retrieving Embeddings from NoRaRe

Thanks to the pynorare API and the CSVW specifications, it is now straightforward to retrieve the different embeddings. Start by creating a fresh virtual environment, install `pynorare` via `pip` and clone the latest version of the Concepticon and NoRaRe datasets:

```
$ python -m venv venv
$ source venv/bin/activate  # for *NIX-based systems; if you use
Windows, you need to run the 'Activate.ps1' script instead
$ git clone --depth=1 https://github.com/concepticon/concepticon-
data
$ git clone --depth=1 https://github.com/concepticon/norare-data
```

Now that you're set up, you can easily retrieve the described embeddings, as illustrated in the code snippet below. Thanks to the CSVW specifications, all data are already represented as objects of the correct type — the embedding vectors are already lists of floats, and the Concepticon ID's are already integers.

```
from pynorare import NoRaRe
from pyconcepticon import Concepticon

# set up Concepticon and NoRaRe API
c = Concepticon("concepticon-data")
norare = NoRaRe("norare-data", concepticon=c)
# retrieve data
embedding_data = norare.datasets["Rubehn-2025-ConceptEmbeddings"]
concept_embeddings = {
   concepticon_id: entry["embeddings_full_affix"] for
   concepticon_id, entry in embedding_data.items()}
ft_embeddings_es = {
    concepticon_id: entry["fasttext_es"] for concepticon_id, entry
   in embedding_data.items()}
```

5 Conclusion and Outlook

In this study, I have briefly illustrated how vector data can be added to and retrieved from NoRaRe by the concrete example of concept embeddings (Rubehn and List 2025) and multilingual word embeddings (Grave et al. 2018). I am optimistic that this addition will be useful for future research representing semantics on a conceptual level, since distributional semantic representations have proven themselves as powerful tools already. Furthermore, the inclusion of embeddings serves as yet another template for how complex data types can be handled by NoRaRe (Ahmedović 2025).

References

Ahmedović, Mira (2025): Handling Non-Standard Datasets in NoRaRe: A Practical Guide. Computer-Assisted Language Comparison in Practice 8.1. 17–23. https://doi.org/10.15475/calcip.2025.1.3

Bojanowski, Piotr and Edouard Grave and Armand Joulin and Tomas Mikolov (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5. 135-146. https://doi.org/10.1162/tacl_a_00051

Forkel, Robert and Johann-Mattis List (2024). PyNoRaRe [Python package, version 1.1.0]. https://pypi.org/project/pynorare/

Gower, Robin (2021): CSV on the Web. Stirling: Swirrl. https://csvw.org

Grave, Edouard and Piotr Bojanowski and Prakhar Gupta and Armand Joulin and Tomas Mikolov (2018). Learning word vectors for 157 languages. In Proceedings of the Eleventh International Conference on Language Resources and Evolution (LREC 2018). https://aclanthology.org/L18-1550/

Gray, Robert M. and David L. Neuhoff (1998). Quantization. IEEE Transactions on Information Theory, 44.6. 2325-2383. https://doi.org/10.1109/18.720541

List, Johann-Mattis (2021). Mapping Multi-SimLex to Concepticon. Computer-Assisted Language Comparison in Practice 4.3. 1-8. https://doi.org/10.58079/m6l1

List, Johann-Mattis (2025). Illustrating Data Curation in NoRaRe with the Help of Templates. Computer-Assisted Language Comparison in Practice 8.2. https://doi.org/10.15475/calcip.2025.2.2

List, Johann Mattis and Annika Tjuka and Frederic Blum and Alžběta Kučerová and Carlos Barrientos Ugarte and Christoph Rzymski and Simon Greenhill and Robert Forkel (eds.) (2025). CLLD Concepticon 3.4.0 [Data set]. Zenodo. https://doi.org/10.5281/zenodo.14923561

Mikolov, Tomas and Kai Chen and Greg Corrado and Jeffrey Dean (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

Pennington, Jeffrey and Richard Socher and Christopher D. Manning (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1532-1543. https://doi.org/10.3115/v1/D14-1162

Rubehn, Arne and Johann-Mattis List (2025). Partial colexifications improve concept embeddings. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 20571-20586. https://aclanthology.org/2025.acl-long.1004

Tjuka, Annika and Robert Forkel and Johann-Mattis List (2022). Linking Norms, Ratings, and Relations of Words and Concepts Across Multiple Language Varieties. Behavior Research Methods 54. 864–884. https://doi.org/10.3758/s13428-021-01650-1

Vulić, Ivan and Simon Baker and Edoardo Maria Ponti and Ulla Petti and Ira Leviant and Kelly Wing and Olga Majewska and Eden Bar and Matt Malone and Thierry Poibeau and Roi Reichart and Anna Korhonen (2020): Multi-SimLex: A large-scale evaluation of multilingual and cross-lingual lexical semantic similarity. Computational Linguistics 46.4. 847-897. https://doi.org/10.1162/coli_a_00391

Supplementary Material

The data described here is available as part of the NoRaRe database, which is currently curated on GitHub (https://github.com/concepticon/norare-data) and regularly archived with Zenodo (all versions available at https://doi.org/10.5281/zenodo.3957680). The data created here will be part of the next release Version 1.2. For details, the contribution can be inspected via the pull-request on GitHub at https://github.com/concepticon/norare-data/pull/282.

Funding Information

This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101044282). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.