Neural Approach to Detecting and Solving Morphological Analogies across Languages

UNIVERSITÉ DE LORRAINE
Loria
IDMC Institut des sciences du Digital Management & Cognition
COMPOSANTE DE L'UNIVERSITÉ DE LORRAINE

Safa AlSaidi, Amandine Decker, and Puthineath Lay

Supervisors: Esteban Marquer and Miguel Couceiro

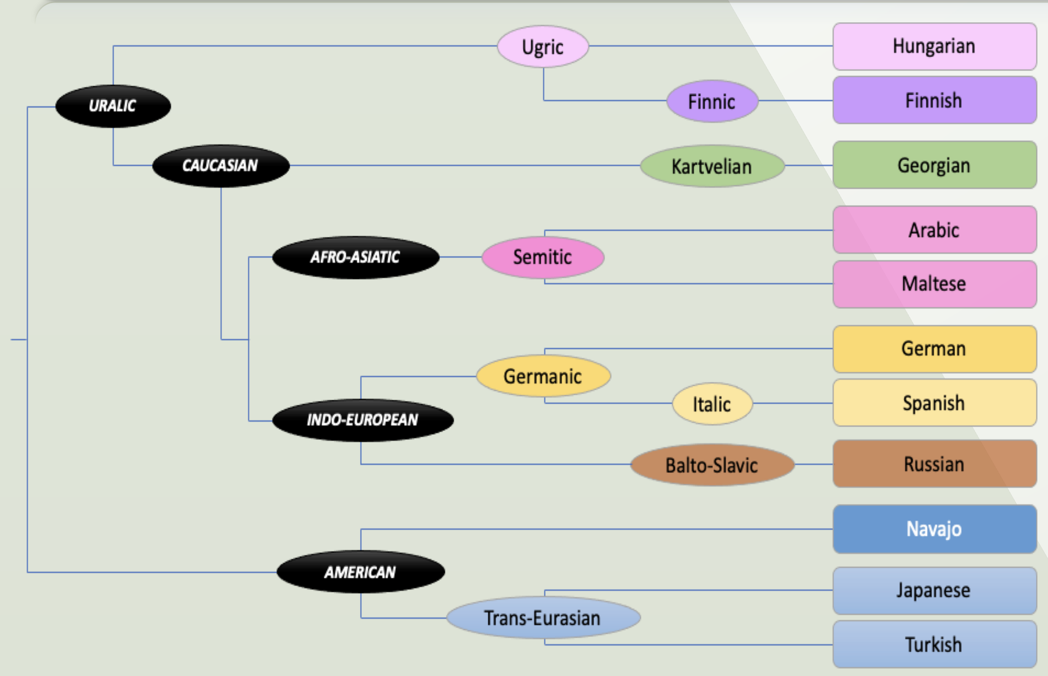
ANALOGIES AND PROJECT OBJECTIVES

Analogies draw a **parallel** between two pairs of words as in "king is to queen what man is to woman". Morphological analogies follow the same principle with morphologically related words as in "cat is to cats what star is to stars". We denote an analogy "A is to B what C is to D" by "A:B::C:D".

In this project we aim **to detect and solve morphological analogies across 11 languages** by:

- building a model that automatically determines if four words form a valid analogy;
- building a model that can solve morphological analogical equations;
- determining whether different languages share morphological properties.

LANGUAGES



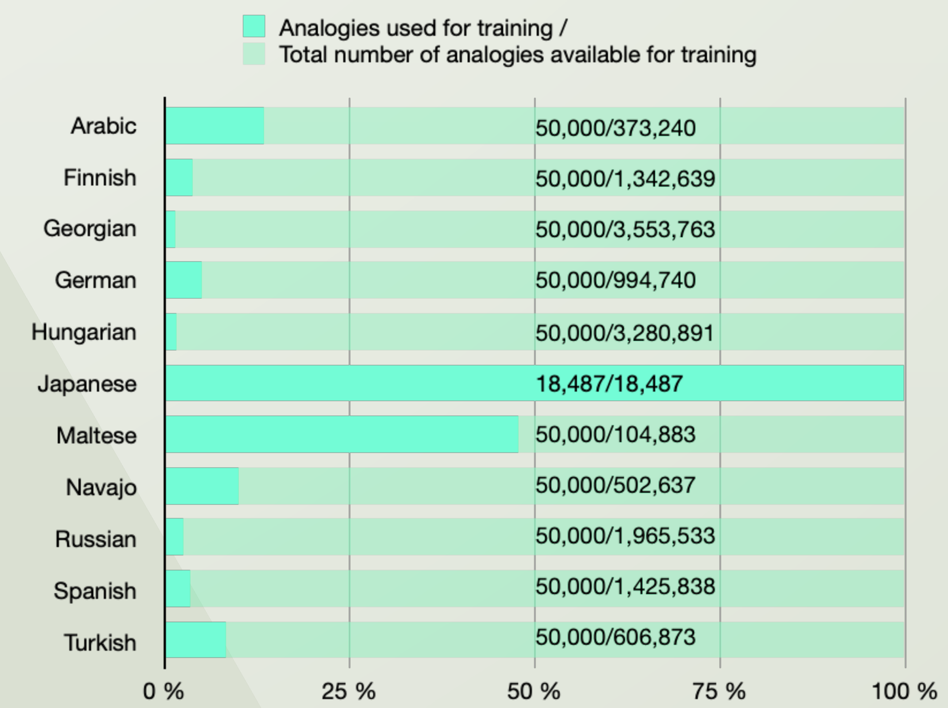
We worked on 11 languages: Hungarian, Finnish, Georgian, Arabic, Maltese, German, Spanish, Russian, Turkish and Japanese.
Some of these languages particularly differ in some way:

- Japanese had the largest number of different characters (632 for <60 for the other languages)
- Japanese, Georgian and Russian do not use the Latin alphabet (Arabic's words are written with the Latin alphabet)
- Maltese is originated from Arabic but underwent the influence of French, Sicilian, Italian and English and thus has a different morphology from Arabic

DATASETS

We used two datasets: SIGMORPHON 2016 (Cotterell *et al.*, 2016) and the Japanese Bigger Analogy Test Set (Karpinska *et al.*, 2018). Our datasets contain triples (lemma, target features, target word) such as (cat; pos=N, num=PL; cats). We generate analogies based on triples sharing the same features. If A:B::C:D is valid then seven permutations are also valid. Below are some invented examples from English:

cat pos=N, num=PL cats
apple pos=N, num=PL apples
→ cat:cats::apple:apples is a valid analogy
→ cat:apple::cats:apples is a valid analogy
→ cat:apples::cats:apple is invalid (wrong form)
→ cat:cat::apple:apples is invalid (wrong form)
cat pos=N, num=PL cats
sleep pos=V, tense=PRS, per=3, num=SG sleeps
→ cat:sleep::cats:sleeps is invalid (not the same features)



We do not use our full datasets for training: **our models are not data voracious !**

REFERENCES

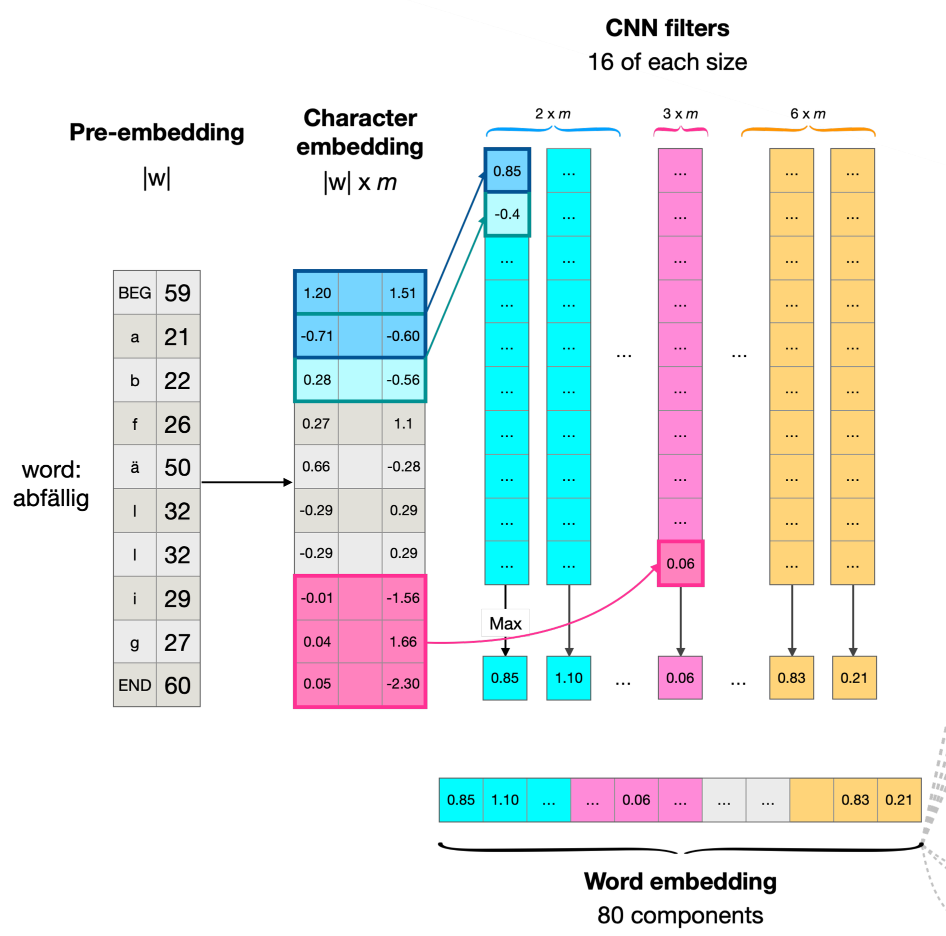
Cotterell *et al.* (2016). The sigmorphon 2016 shared task—morphological reinflection. In the Proceedings of the ACL 2016 Meeting of SIGMORPHON.

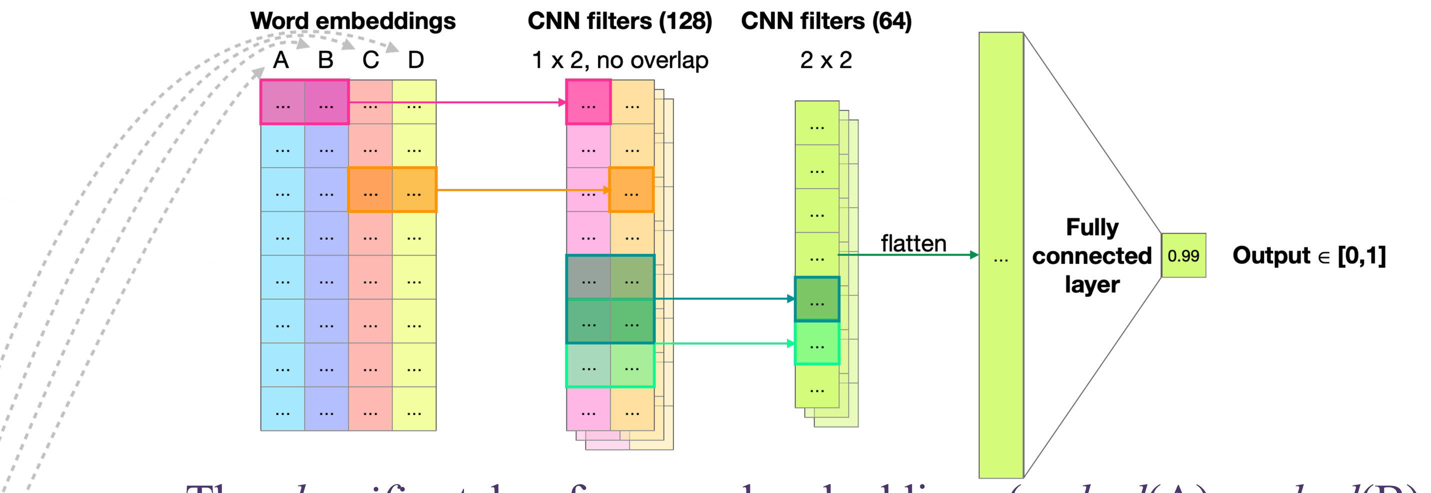
Karpinska *et al.* (2018). Subcharacter Information in Japanese Embeddings: When Is It Worth It? In the Proceedings of the ACL Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP, (pp. 28–37).

Lim *et al.* (2019). Solving word analogies: A machine learning perspective. In the Proceedings of the 15th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU 2019 (pp.238–250).

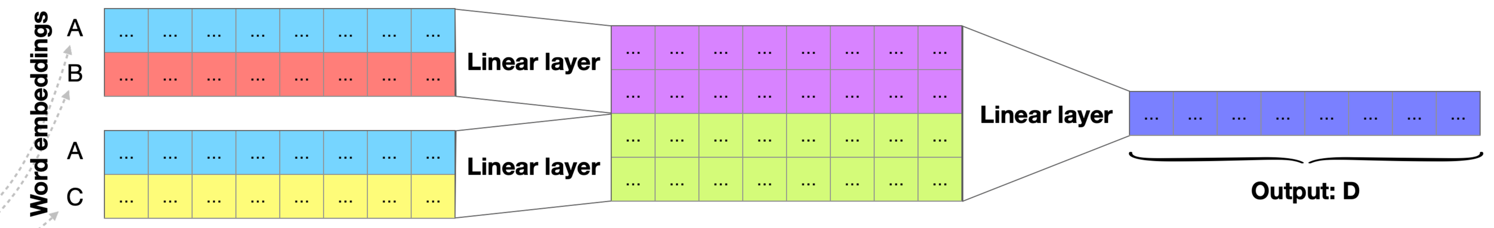
OUR NEURAL NETWORKS …

To work with natural language, we need a way to numerically represent the words: a *word embedding* model (on the left). Additionally, we use one model to classify analogies and another to solve analogical equations (Lim *et al.*, 2019).



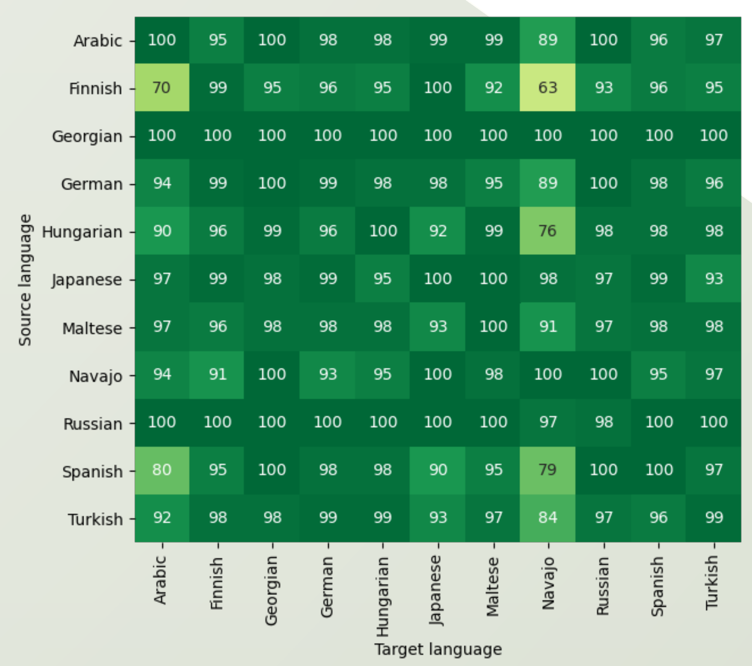


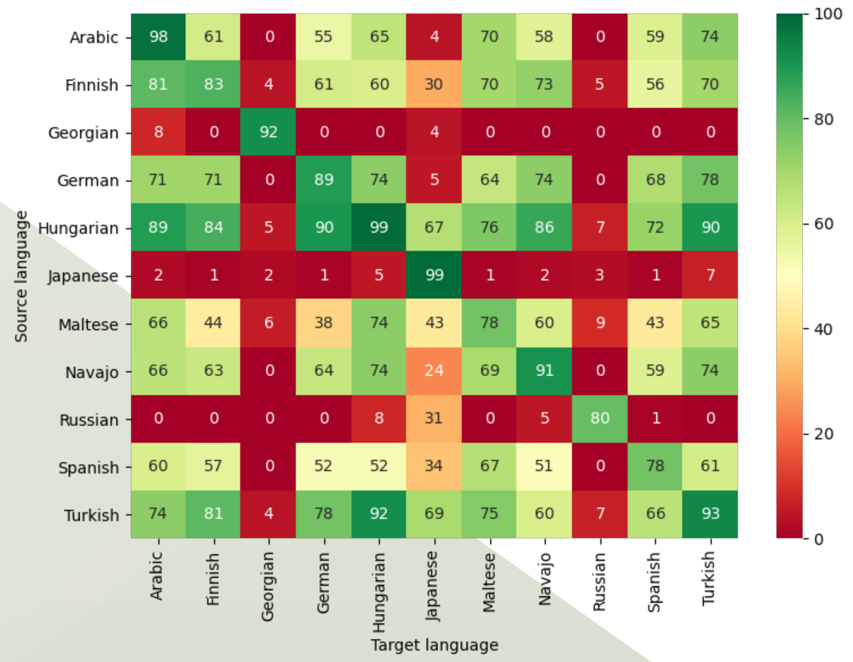
The *classifier* takes four word embeddings (*embed*(A), *embed*(B), *embed*(C), *embed*(D)) as input and outputs a probability for A:B::C:D to be a valid analogy.



The *analogy solver* takes three word embeddings (*embed*(A), *embed*(B), *embed*(C))) as input and outputs a vector which should correspond to *embed*(D) such that A:B::C:D holds true.

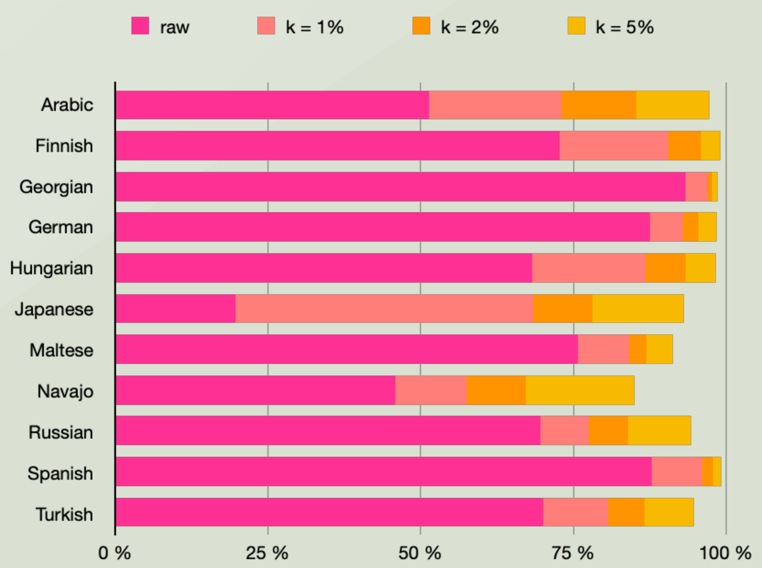
… TO DETECT ANALOGIES ACROSS LANGUAGES …

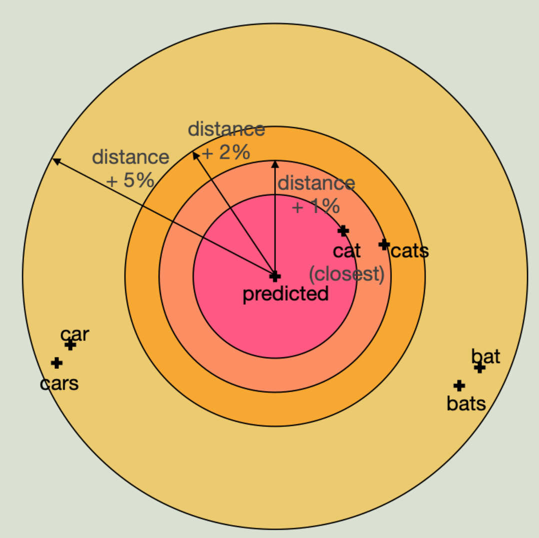




Our first neural network is able to classify quadruples of words as valid or invalid analogies. We trained one model per language and then evaluated each of them on all the languages. The values in the confusion matrices correspond to the portion of valid/invalid analogies classified as valid/invalid.
The results from Georgian, Japanese and Russian are probably due to the fact that the alphabet these languages use are not recognised by the other models.

… AND SOLVE ANALOGICAL EQUATIONS





Our second neural network solves analogical equations: given (A, B, C) it produces D such that A:B::C:D is valid. For instance it should produce "cats" with the input ("star", "stars", "cat").
The model produces numerical representation of words. We search the closest corresponding word among the words of the dataset. If it is the right one, we consider the model was right. This evaluation corresponds to the *raw* values.
We also searched for the right word a bit further. The diagram on the right (not at scale) illustrates the process for $k \in \{0.01, 0.02, 0.05\}$.