Multilingual Base Word Recognition in Derivation

Vojtěch John, Zdeněk Žabokrtský

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

CC BY SA
unless otherwise stated

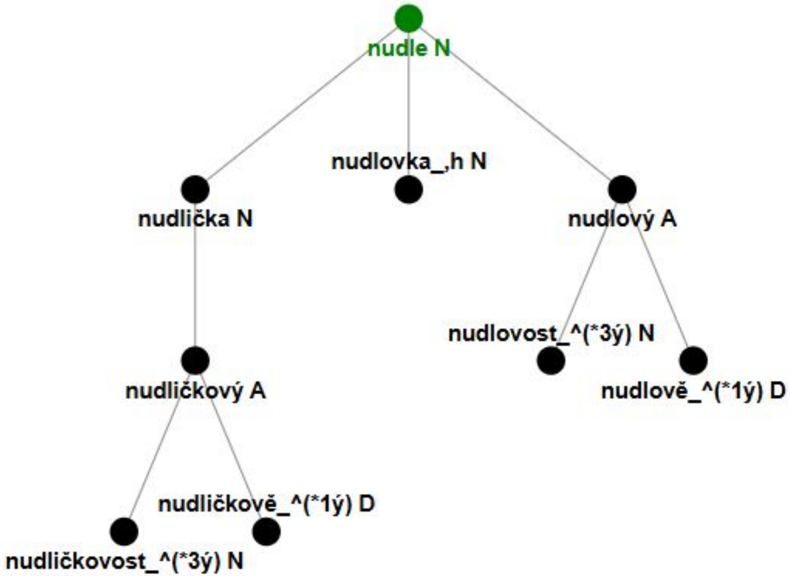
Introduction

Motivation

- Creation of static derivational resources is hard
  - Manual methods are labor- and time- intensive
  - Unsupervised and semi-supervised methods are usually unreliable
- Supervised extension of existing resources
- Dynamic modelling of derivation might be useful
- Predicting base words - we assume derivation is directional
- Candidate base words can be available
  - For given word and set of candidate parents, select the most probable parent
  - We need to decide for a word and candidate base word, whether the candidate is naše word
  - in: “kindly, kind” - out: True
- Candidate base words can also be unavailable
  - For given word, generate base words
  - in: “kindly” - out: “kind”

Data: Universal Derivations

- Largest current derivational resource
- 28 datasets of varying size and quality
- 20 languages in total, mostly Indo-european
- Derivational relations presented as edges of trees (or directed acyclic graphs)
- Some resources include also:
    - Compounding
    - Conversion
    - Word variants



Data: Remarks on preprocessing

- Train-dev-test split on trees (s.t. the overlap between common base words is minimal)
- We extract pairs (*word-base word(s))*
- Negative training samples (and test samples) generated automatically
    - *Word* and *Base word* always taken from the same derivational tree
- We treat the datasets as gold data
    - Missing edges
    - Incorrect or debateable edges
    - Design decisions

Experiment 1: Selecting base words

Task formulation and data

- Given a pair (*candidate base word/s, child),* decide whether such a pair constitutes a valid derivational relation or not.
    - Input: (kind, kindly)
    - Output: True
- We train binary classifiers
- Training on each data resource separately
- As test data, we take 5 % of the total data
- Negative examples
    - Sampled from words present in the same derivational tree
    - Approx. the same amount of positive and negative examples in each dataset
    - The numbers Is arbitrary

Classifiers - ablation study

- **Neural networks**
  - Words in a fixed frame, forwards and backwards (e.g. “*[e, g, g, 0, 0, ..., 0, g, g, e]*”)
  - Two classification heads (*Parent* and additional *Relative*) with dense layers.
- ***Simple***
  - ***Inputs***: Product of *fasttext* embeddings, Levenshtein distance
- ***Cosine***
  - ***Inputs***: cosine distance of fasttext embeddings, candidate word pair (*word, base word)* words (processed by ResNet blocks)
- ***Full***
  - ***Inputs***: the two words and their fastText embeddings
  - words are embedded and processed by ResNet blocks
  - Embeddings: a dense layer with dropout, multiplied and then a convolutional layer
- ***Subtract***
  - ***Inputs***: Difference between fasttext embeddings, difference between words; otherwise same as in full

Results

Setting	Binary accuracy	Precision	Recall
Cosine	88.81 %	74.42 %	77.43 %
Simple	78.91 %	46.92 %	51.24 %
Subtract	***93.05 %***	***88.24 %***	***83.87 %***
Full	87.45 %	73.26 %	79.05 %

- ***Subtract*** being best corresponds to a finding by (Musil et al., 2019) - differences of word embeddings ~ meanings of derivational affixes

Final version

- Modified version of ***Subtract*** (e.g. embedding difference is fed to a transformer decoder block)
- Macroaverage across datasets:
  - Precision 91.2 %
  - Recall 90.8 %
- Effect of dataset size small if any
- Effect of data quality seems much larger

Experiment 2: Generating base words

Methods

- Neural networks with Transformer architecture
- Monolingual models with ablations
  - *Basic* (small transformer - 2 layers)
  - *Big* (increase size)
  - *BPEmb* (add BPEmb embeddings to the input)
  - *FastText* (add FastText embeddings to the input)
  - Early stopping
- Fine-tuned multilingual models
  - ByT5 models (no tokenization)
  - Small and Base versions
  - Finetuned on all the datasets together
  - 5 epochs

Data

- 500 trees for test data, 100 for validation data
  - if not available, 50 % of trees to train set, 10 % to validation set
- Multilingual models - combined resources
  - Same language, different resources
  - Throw away overlaps of train and test
  - No effort to resolve inconsistencies
    - E.g. missing edges in one of the resources
  - May unfairly improve the performance over small test sets

Results

- Metrics: Word-level precision
- Results vary wildly across resources
    - Size helps, but quality helps more
- Finetuned models perform best
- Model size does not seem to matter
- FastText embeddings help, BPEmb embeddings don’t.
    - Perhaps the models want morphological information, not semantics?

	Basic	Big	BPEmb	FastText	ByT5-small	ByT-basic
Macro Average	62.3	61.3	59.2	64.0	68.0	68.0

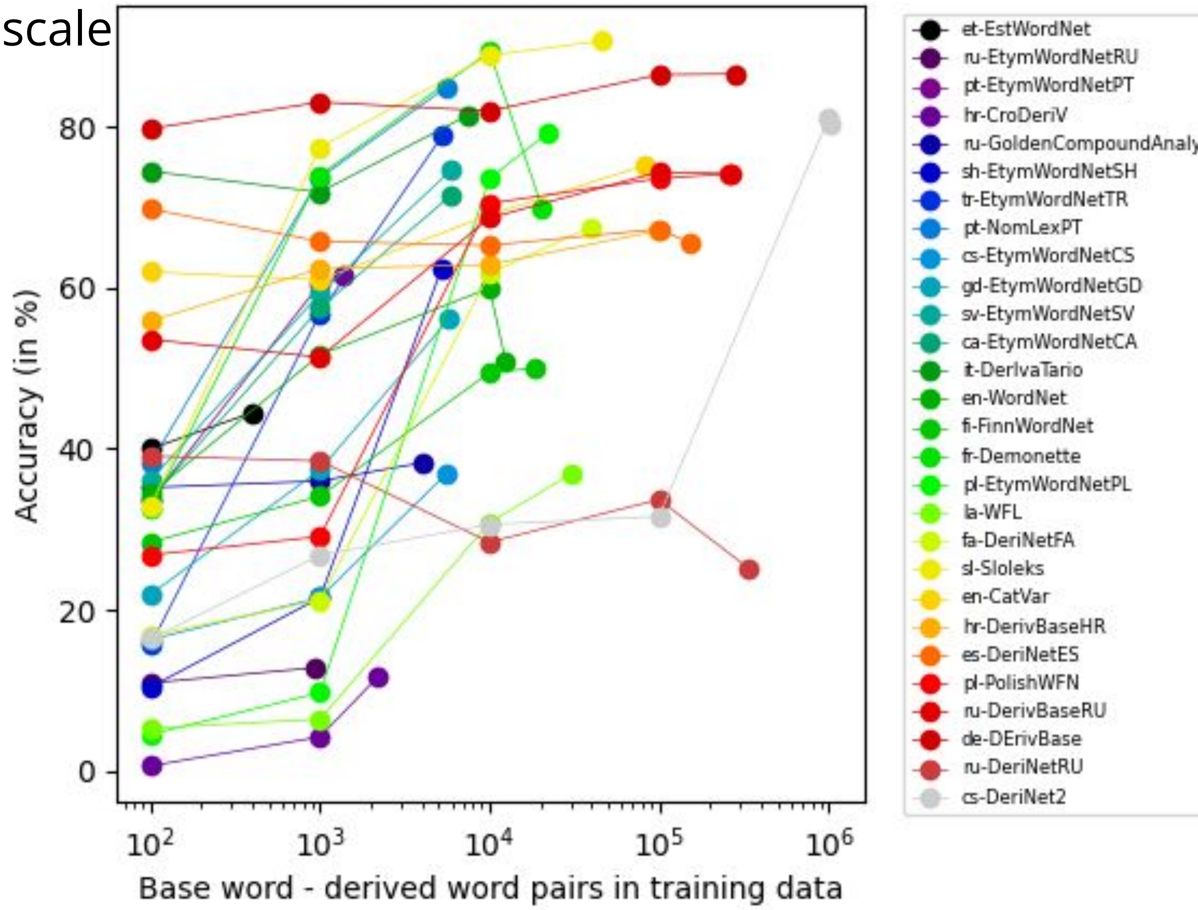
Binned macroaverage

- We binned the results (4 bins, 7 results each)
- FastText and ByT5 models are more robust than the rest
  - No observable effects of the curse of multilinguality

Bins	4	3	2	1
Basic	30.5	60.7	74.1	83.8
Big	30.3	59.1	72.5	83.1
BPEmb	22.7	58.9	72.7	82.5
FastText	34.8	62.6	74.1	84.3
ByT5-small	35.5	66.6	**80.3**	**89.5**
ByT5-basic	**36.0**	**66.6**	80.2	89.4

(FastText) learning curves are a mess

- First experiment on this scale
- Initial points
    - (Resource) complexity
- Flat curves
    - Simple resources
    - Automatic generation
    - Reverse engineering
- Steep curves
    - When?
    - (Language) complexity
- Simple resources
    - 1000 to 10,000 examples
- Difficult resources
    - Over 100,000 examples?



Interesting error types

- Wrong order of word-formation operations
  - *Overwhelming - *whelming*
- More or less plausible but non-existent base words
  - *Západopennsylvánský - *západopennsylván* (*west-pennsylvanian - *west-pennsylvan)*
  - *Svatba - *svat* (n.b. etymologically correct)
  - *Antibióza - *bióza (antibiosis - *biosis)*
- Conversion resolution
  - Is “festering” (NOUN) a child of “fester” or “festering” (VERB)?
  - Possible solution: add POS tags
- Word variants
  - *Oučinkování - účinkování* vs *oučinkování - oučinkovat*

Summary

- We have trained state-of-the-art models for base-word identification & generation
- Training data quality is crucial
- Simple vs complex resources
- Multilingualityimproves robustness
- **Thank you for your attention!**