A filter for syntactically comparable parallel sentences

Martin Kroon

23 October 2018

My project

automatic detection of syntactic differences between languages

Why?

- comparative syntax
  - finding a universal grammar that underlies all natural languages
- automation means:
  - more data
  - faster
  - unbiased

Micro or macro

- micro:
  - you need parallel data
  - you know where the differences in sytax occur
- macro:
  - unparallel data possible
  - then you won't know **where** the differences occur
  - only difference in characterizing patterns

Europarl corpus

- proceedings of EU parliament, translated in all languages
- parallel corpus with multiple languages
- about 2 million sentences per language
- quite **noisy**!

Filter

we need to clean up the data

- wrongly aligned sentences
- 'free' translations

Filter

4 filters

- baseline: Levenshtein distance on POS tags
- sentence-length ratio
- Levenshtein further explored
- graph-edit distance on dependency parses
- sentence vectors

Syntactic comparability

That is what will make us strong.	Dan zijn wij sterk.
... I hope that this report will not be allowed **to bite the dust** on account of this...	... hoffe ich, dass dieser Bericht nicht deswegen **zu Fall gebricht wird**...
This can double the available resources.	Hierdoor kunnen de beschikbare middelen worden verdubbeld.
The house was destroyed by Jim.	Jim tuhosi talon.

Data

manually annotated 250 English-Dutch sentence pairs

- 105 comparable
- 145 not comparable

three annotators $\kappa = 0.70$

If:

- all content words in sentence A have an alignment with a word in sentence B and all content words in sentence B have an alignment with a word in sentence A, where punctuation is to be ignored
- if there is no paradigm shift, such as active to passive, idiomatic constructions in one language or a (pseudo-)cleft in one language, ignoring word order

Universal Dependencies

Dependency trees are a way of representing syntactic structure in a sentence, where child nodes 'depend' on their mothers.

Universal Dependencies is a programme that aims for cross-linguistically consistent tagging and annotation of dependency trees.

- We tagged and parsed our data in UDPipe

Back to filters

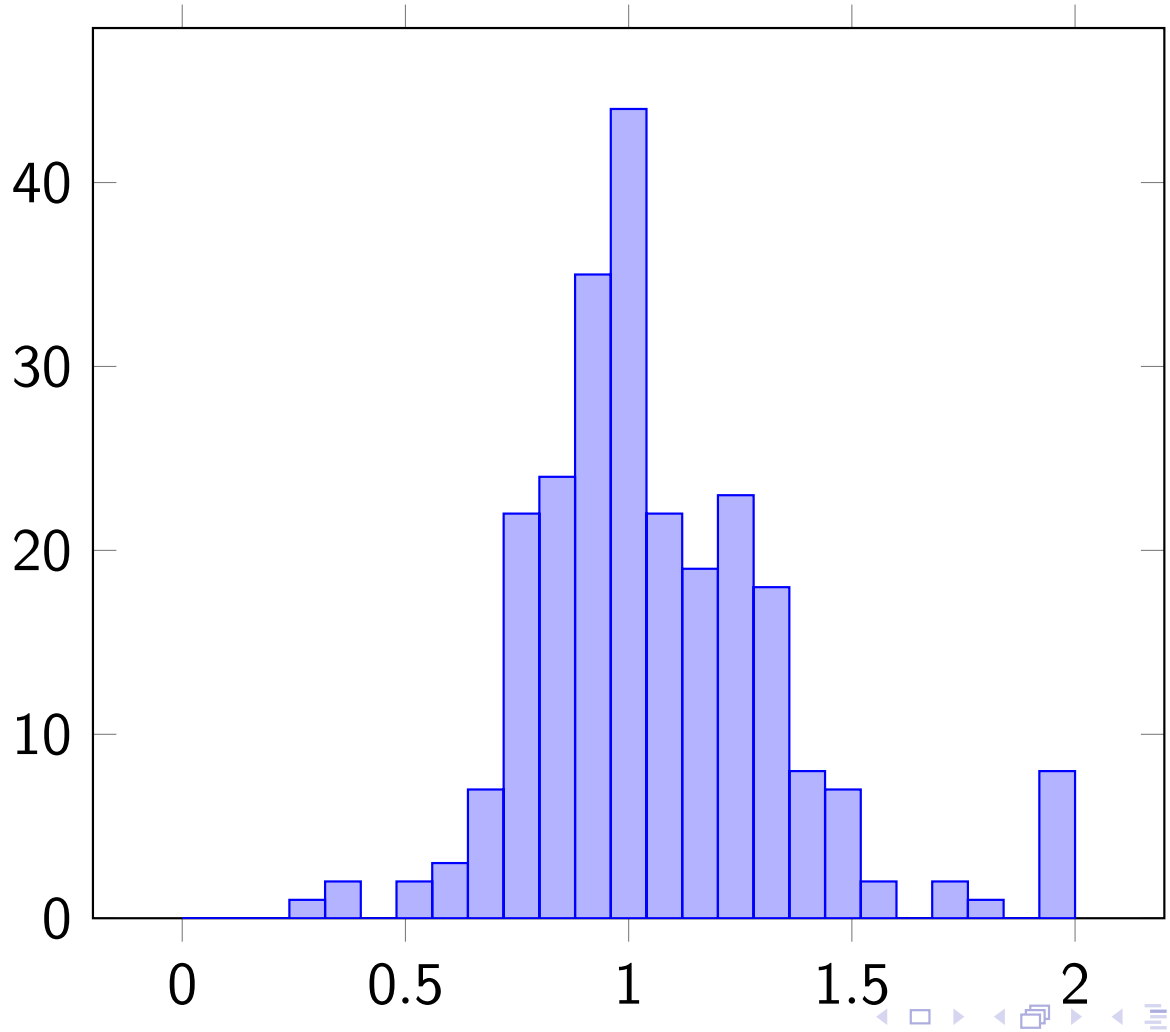
baseline:

- Levenshtein distance on POS tags
- if edit distance higher than certain threshold, discard sentence pair
- thresholds determined with an ROC curve

First filter

sentence-length ratio

- relative sentence length
- if ratio too high or too low, discard sentence pair
- in terms of percentiles



First filter

sentence-length ratio, cont'd

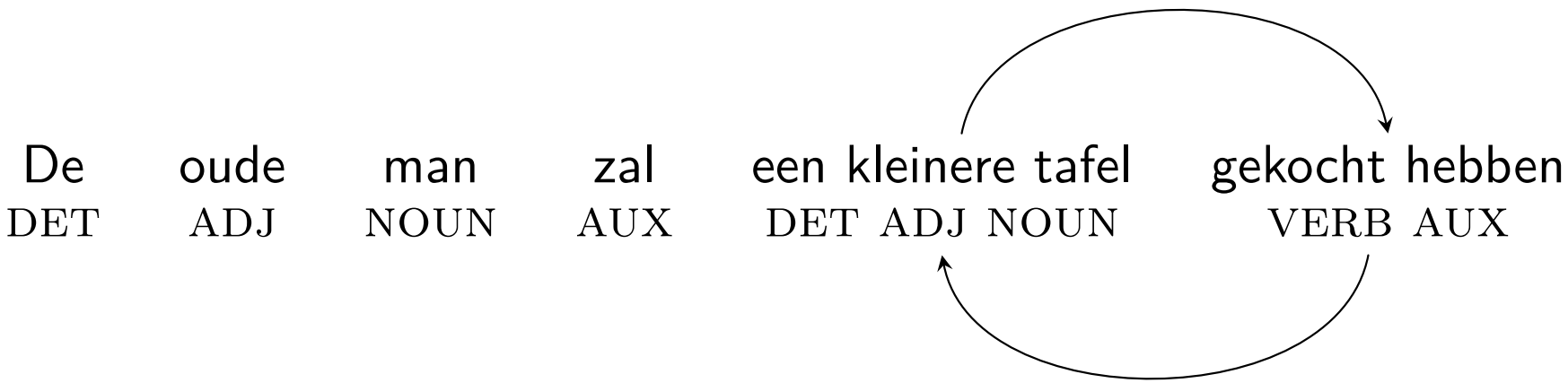
- experimented with ignoring function words
- concern: coarse-grained

Second filter

Levenshtein further explored

- experimented with ignoring function words
- transpositions
- concern: sensitive to constituents transposing

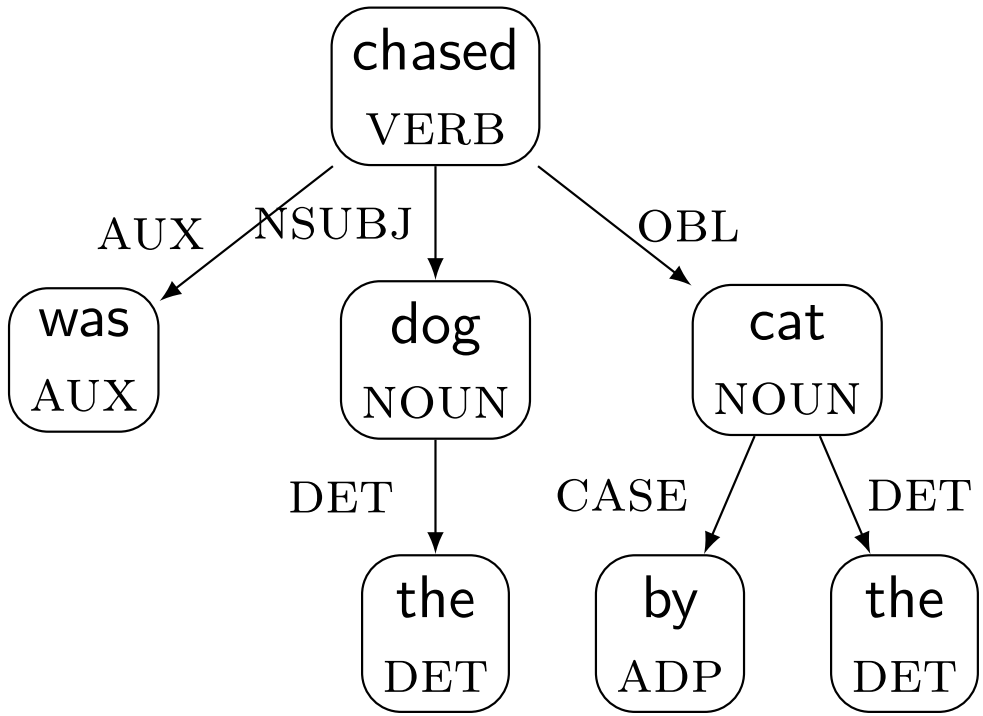
The	old	man	will	have bought	a smaller table
DET	ADJ	NOUN	AUX	AUX VERB	DET ADJ NOUN



Third filter

graph edit distance

- edit distance on dependency parses as graphs
  - `networkx`
  - node and edge identity in terms of POS and syntactic relation
  - insertion, deletion, substitution $= 1$
  - unordered graphs
- if edit distance higher than certain threshold, discard sentence pair



Third filter

GED cont'd

- experimented with including morphological information
- experimented with ignoring function words
- concern: very reliant on parse accuracy, requires existence of parser

Fourth filter

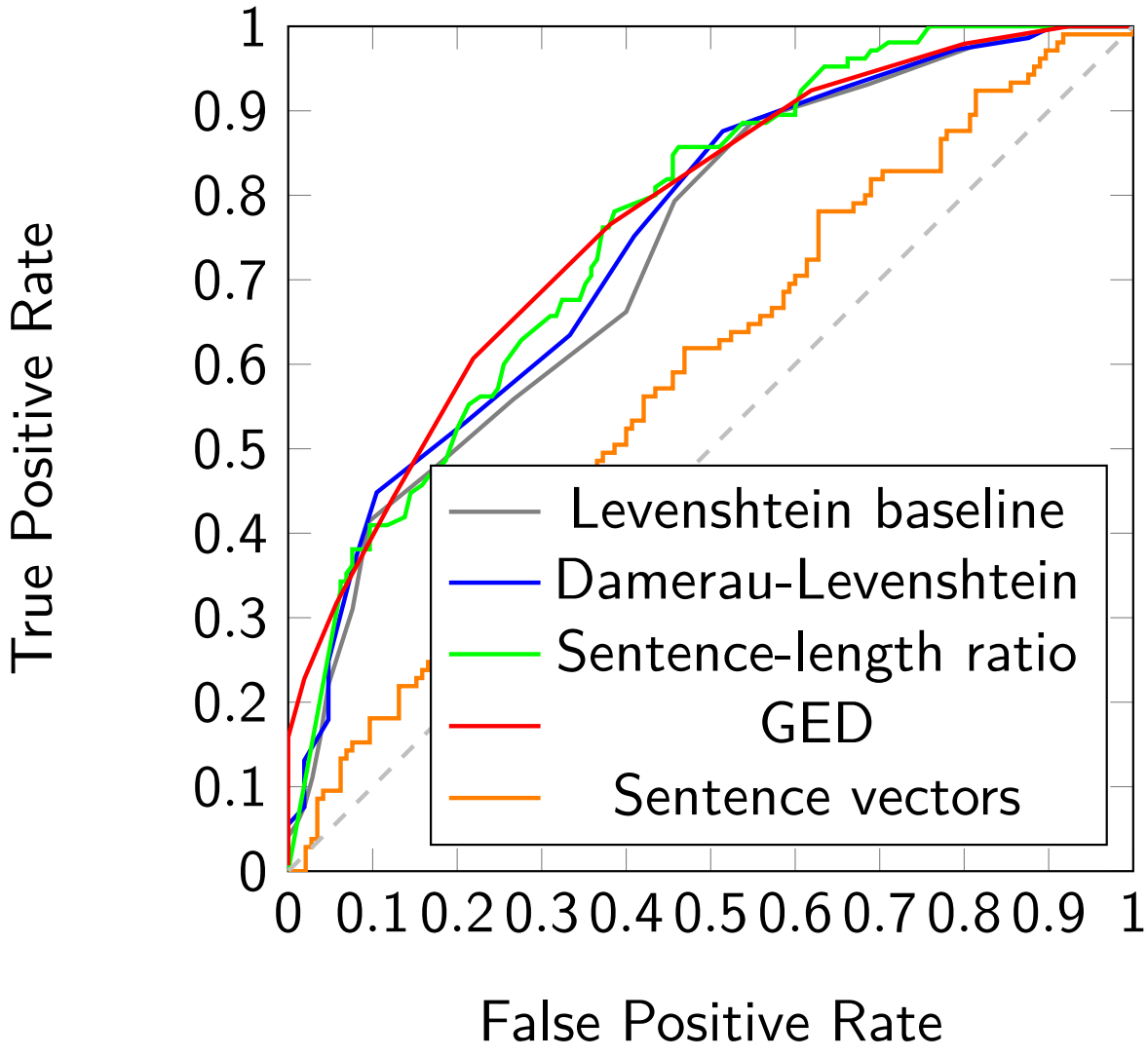
sentence vectors

- cosine similarity between sentence vectors
  - average of all word vectors
  - required translation matrix à la Mikolov 2013
- if cosine similarity below certain threshold, discard sentence pair
- trained vector spaces on Europarl with `fastText`
- experimented with ignoring function words
- experimented with weighting sentence vectors with tf-idf of word vectors

Results

	AUC	thresh.	prec.	rec.	$F_{0.5}$
Baseline	0.73	6	0.72	0.67	**0.71**
GED	**0.77**	3	**0.73**	0.65	**0.71**
Levenshtein	0.75	6	0.72	0.68	**0.71**
Sentence-length ratio	0.76	18.6%	0.71	**0.70**	0.70
Sentence vectors	0.59	0.793	0.58	0.57	0.58

ROC graph


True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Levenshtein baseline
Damerau-Levenshtein
Sentence-length ratio
GED
Sentence vectors

Problems

- only English-Dutch
  - English-Danish
  - English-...
- small data set
  - probably going to expand it
  - manual POS tagging and parsing?
- maybe combining the filters, but have to look into that
- unsupervised filter training/setting of threshold???