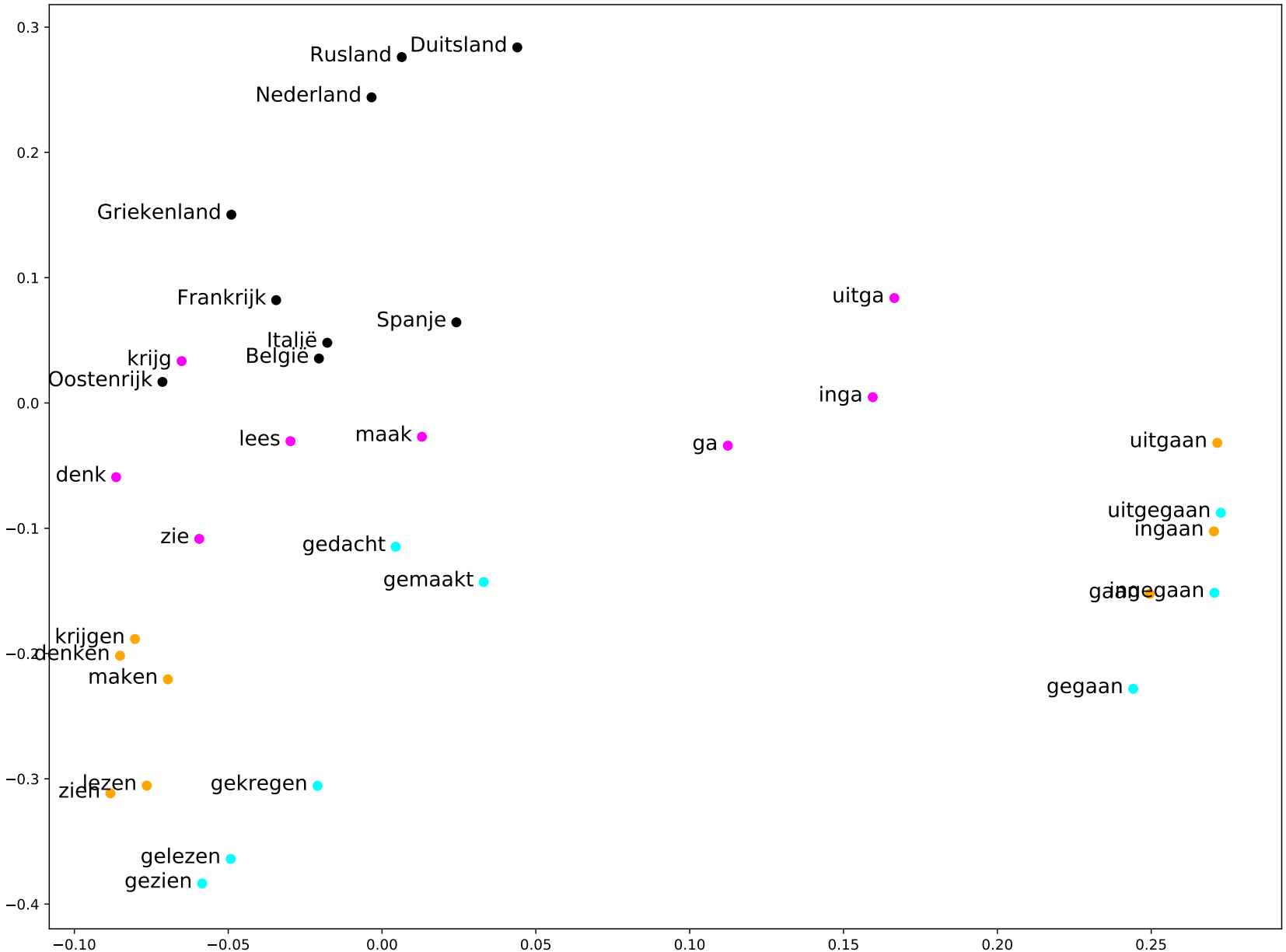
Dutch Spelling Correction via KPCA Embeddings

Eduardo Brito, Rafet Sifa, Christian Bauckhage

Fraunhofer Institute for
Intelligent Analysis and Information Systems IAIS

26th January 2018

KPCA Embeddings: Toy Vocabulary Example



Language Model: Approach

Represent words as vectors

KPCA embeddings[a]

---

[a] E. Brito, R. Sifa, C. Bauckhage. KPCA Embeddings: an Unsupervised Approach to Learn Vector Representations of Finite Domain Sequences. In *Lernen, Wissen, Daten, Analysen* , 2017.

Words with "similar" form close in the vector space

Bigram similarity[a]

---

[a] G. Kondrak. N-gram similarity and distance. In *String processing and information retrieval*. Springer Berlin/Heidelberg, 2005.

Vocabulary: Dutch word list

wdutch (available as Debian package, OpenTaal Project)

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling
4. Correct the misspelling getting the nearest word vector, unless the word token...

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling
4. Correct the misspelling getting the nearest word vector, unless the word token...
   - is a number
   - ends with a dot (likely an abbreviation)
   - it does not contain no other special characters but "-" and "'"
   - is just punctuation ("."."!", "?", "...")

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling
4. Correct the misspelling getting the nearest word vector, unless the word token...
   - is a number
   - ends with a dot (likely an abbreviation)
   - it does not contain no other special characters but "-" and "'"
   - is just punctuation ("."."!", "?", "...")
5. Uppercase only if:
   - Uppercased in the word list
   - Beginning of the sentence

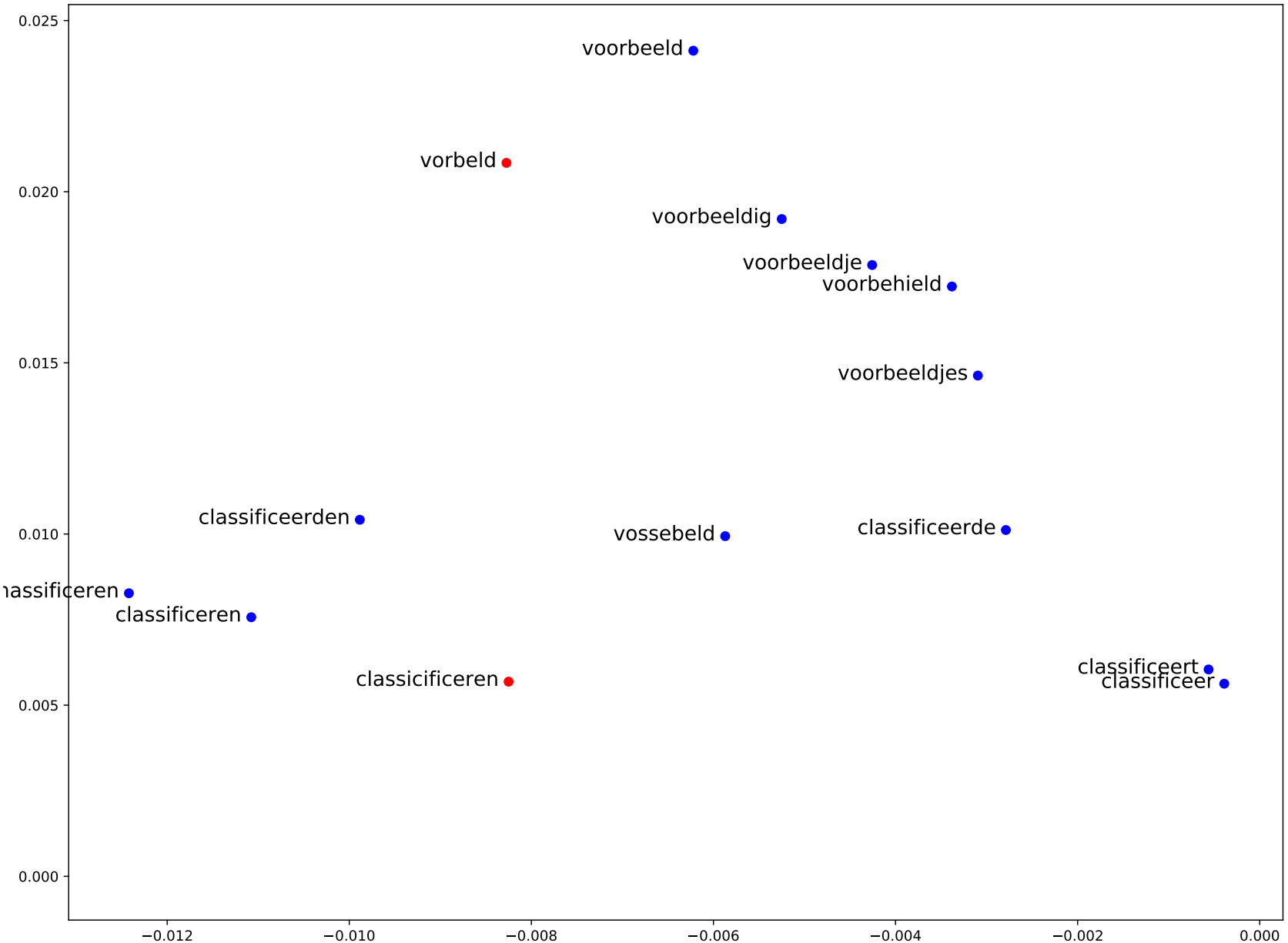
Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling
4. Correct the misspelling getting the nearest word vector, unless the word token...
   - is a number
   - ends with a dot (likely an abbreviation)
   - it does not contain no other special characters but "-" and "'"
   - is just punctuation ("."," !", "?", "...")
5. Uppercase only if:
   - Uppercased in the word list
   - Beginning of the sentence
6. Discard also any correction if:

Spelling Correction Approach

1. Generate a vector representation for all vocabulary words
2. Words not occurring in the word list considered as misspellings
3. Generate a vector representation for each misspelling
4. Correct the misspelling getting the nearest word vector, unless the word token...
   - is a number
   - ends with a dot (likely an abbreviation)
   - it does not contain no other special characters but "-" and "'"
   - is just punctuation ("."," !", "?", "...")
5. Uppercase only if:
   - Uppercased in the word list
   - Beginning of the sentence
6. Discard also any correction if:
   - a run-on error is detected (validity check on all possible splits)
   - nearest KPCA embeddings is "too far" (potential proper noun)

KPCA Embeddings with Bigram Similarity



Results

Recall

- Detection: 0.44 (+0.26)
- Correction: 0.30 (+0.29)

Detection and correction significantly better than baseline ☺

Results

Recall

- Detection: 0.44 (+0.26)
- Correction: 0.30 (+0.29)

Detection and correction significantly better than baseline ☺

Precision

- Detection: 0.07 (-0.37)
- Correction: 0.05 (-0.37)

Too many false positives ☹

Limitations

No contextual information

- Only good at error categories depending on the word form
- Only few rules as workaround
- Many false positives because of capitalization!

Limitations

No contextual information

- Only good at error categories depending on the word form
- Only few rules as workaround
- Many false positives because of capitalization!

Detection limited by the word list selection

- Issues with many compound/inflected/foreign words
- Yet more false positives!

Thank you!

Code available at Github

`https://github.com/ebritoc/kpca_embeddings`

Questions?

eduardo.alfredo.brito.chacon@iais.fraunhofer.de