Towards Domain Adaptation for Dutch Social Media Text Through Normalization

Rob van der Goot, Gertjan van noord
University of Groningen
r.van.der.goot@rug.nl

10-02-2017

Task

- Lexical normalization

Task

- Lexical normalization
- No word reordering

Task

- Lexical normalization
- No word reordering
- But includes multi-word replacements

Task

Why?

Data

- De Clerq, O., Schulz, S., Desmet, B. & Hoste V. (2014). *Towards shared datasets for normalization research.* Proceedings of LREC 2014
- 962 Tweets
- 578 train / 192 dev / 192 test
- Includes multi-word normalization
- Flemish

Data

Sentences	962
Words	12,900
% Words normed	4.80
% Words split	1.07

Data

Example:

Maria	is	deze	week	veeeel	beter	amai	!	Goeie
Maria	is	deze	week	veel	beter	amai	!	Goede

songkeuze	ook	!	Goe	gezonge	ze	maske	_	#tvvv
songkeuze	ook	!	Goed	gezongen	ze	meisje	_	#tvvv

Data

lap	,	bijna	mijnen	Duvel	de	grond	op	.	Azo	ne	rug
lap	,	bijna	mijn	Duvel	de	grond	op	.	Zo	een	rug

kindj
kindje

Data

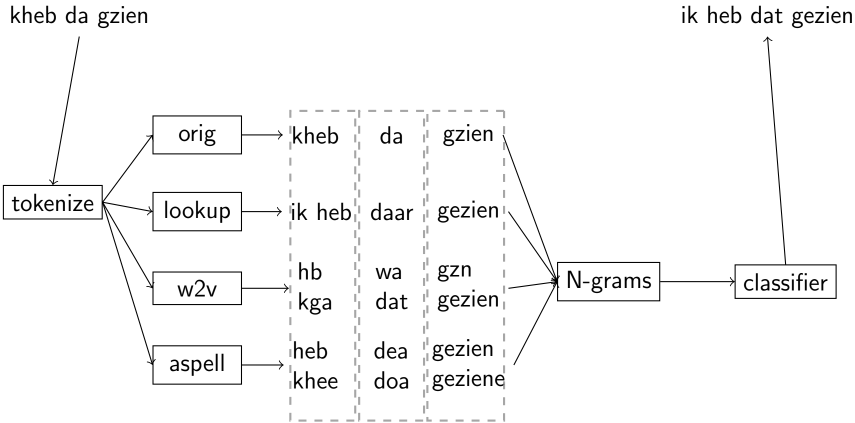
Unigram ranking:

	Normalization	Twitter	Google
1	#tvvv	USERNAME	de
2	de	ik	van
3	een	je	en
4	is	de	in
5	ik	een	een
6	dat	en	het
7	het	het	op
8	in	is	is
9	niet	niet	voor

Data

Deze coaches volgen enkel hun gehoor en gaan met de beste stemmen aan de slag!
the Voice van Vlaanderen
VRIJDAG 21.00
vtm
vtm.be

method



kheb da gzien

tokenize

orig

lookup

w2v

aspell

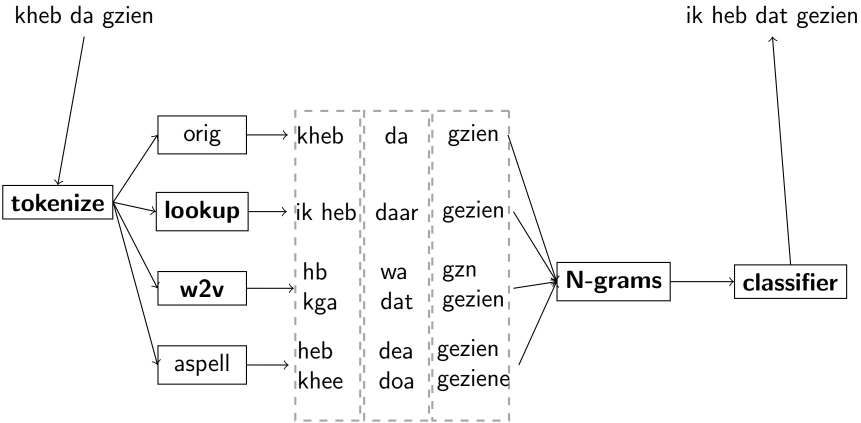
kheb	da	gzien
ik heb	daar	gezien
hb kga	wa dat	gzn gezien
heb khee	dea doa	gezien geziene

N-grams

classifier

ik heb dat gezien

method



kheb da gzien

ik heb dat gezien

tokenize

orig

lookup

w2v

aspell

kheb da gzien

ik heb daar gezien

hb kga wa dat gzn gezien

heb khee dea doa gezien geziene

N-grams

classifier

method

Tokenization:

- Rule based

method

Tokenization:

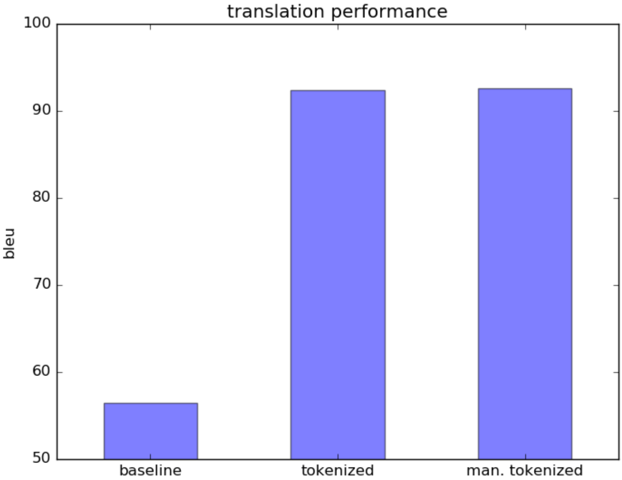
- Rule based
- Split (sequences of) special characters attached to words

method

Tokenization:

- Rule based
- Split (sequences of) special characters attached to words
- Does it work?

method


translation performance
bleu
100
90
80
70
60
50
baseline
tokenized
man. tokenized

Generation

Lookup

- Static lookup dictionary
- Learned from training data
- Only multi-word generation module

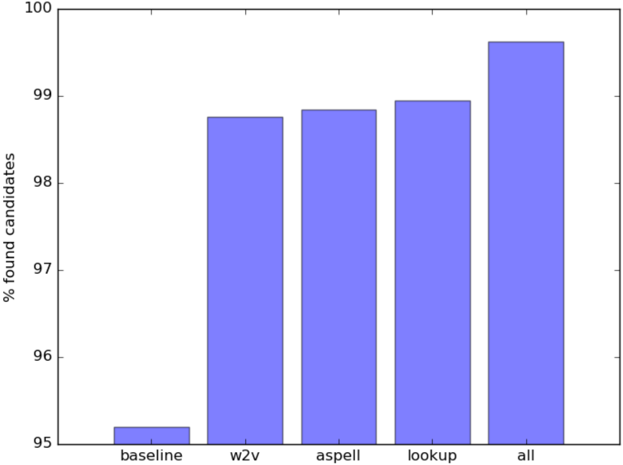
Generation

word2vec

- Trained on 1,545,871,819 tweets
- Vocab size: 5,673,372
- Settings: -cbow 0 -size 400 -window 1 -negative 5 -sample 1e-4 -iter 5

Generation

Generation performance (accuracy)



Generation

Not found normalizations:

harreej	hoorray
ofwat	of wat
koenwauters	koen wauters
h	h
hahahahahahaha	haha

15 cases in train + dev data

Ranking

Additional features

- Dictionary lookup (aspell)
- Order of characters: s.*r.*c.* ⇒ source
- N-grams

Ranking

N-grams

- 1. trained on 1,545,871,819 tweets
- 2. Dutch google N-grams
- Unigram and Bigram probabilities

Ranking

Random Forest Classifier

- Ranger
- Binary classification: task is to predict which word belongs to the corect class.
- Use confidence score to rank.
- 500 trees

Evaluation

Which evaluation metric?

- F1
- WER/accuracy (sometimes with gold error detection)
- bleu

Evaluation

English

- **F1**
- WER/**accuracy (sometimes with gold error detection)**
- bleu

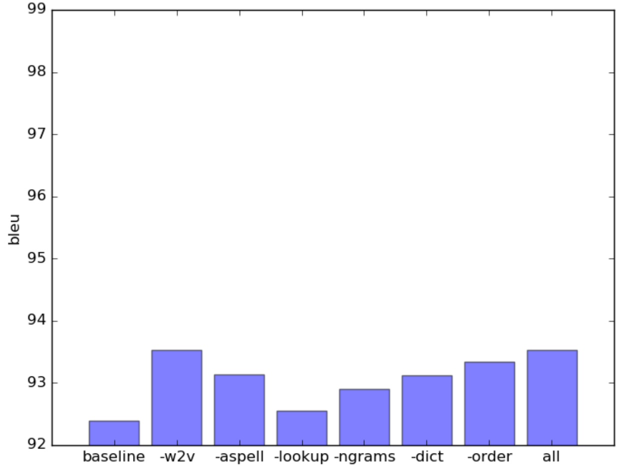
Evaluation

Dutch

- F1
- **WER**/accuracy (sometimes with gold error detection)
- **bleu**

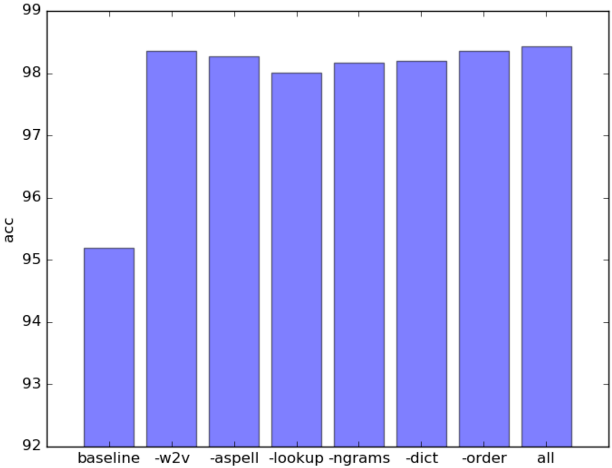
Evaluation

BLEU



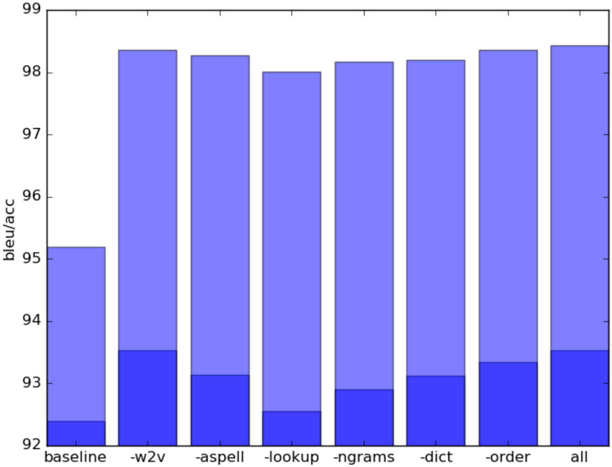
Evaluation

Accuracy



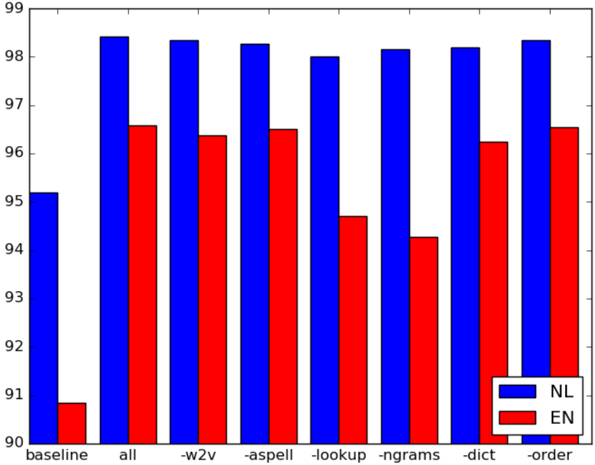
Evaluation

Bleu vs. Accuracy



Evaluation

Feature importance compared to English



Evaluation

Wrongly ranked:

orig	mine	gold
ok	ok	ok
da	dat	de
heul	heul	heel
gister	gister	gisteren
cava	cava	ça va

Conclusion

- N-grams are an important feature for ranking
- A random forest classifier works well for ranking
- Word embeddings are not very useful for normalization, a simple lookup list is
- Bleu vs accuracy (tokenization)

Conclusion

- https://bitbucket.org/robvanderg/monoise