

# Parser Adaptation for Social Media by Integrating Normalization

Rob van der Goot & Gertjan van Noord  
r.van.der.goot@rug.nl

03-03-2017

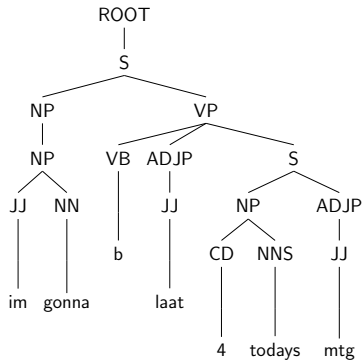
# Problem



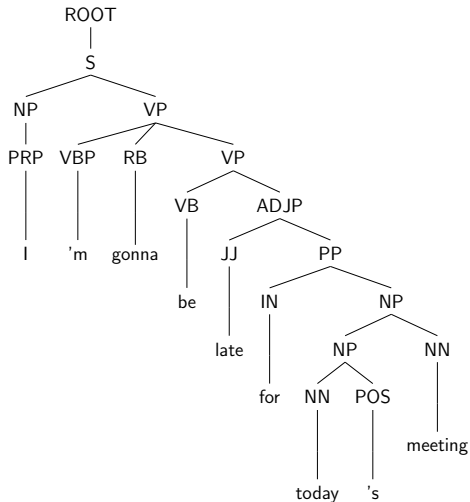
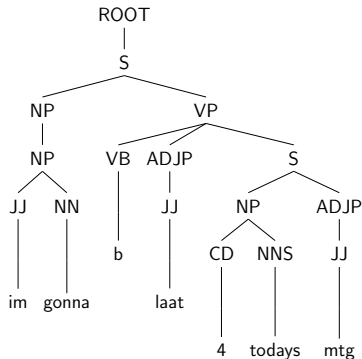
# Problem



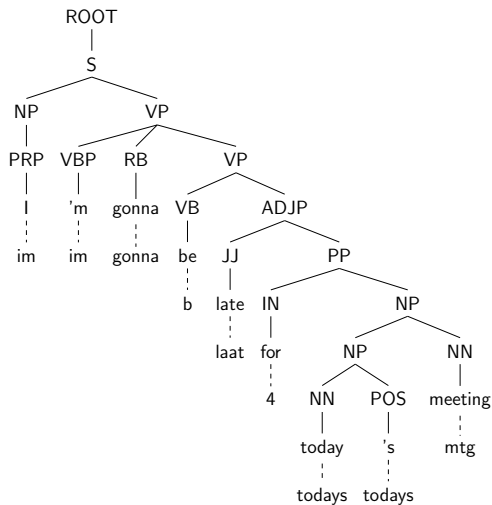
# Problem



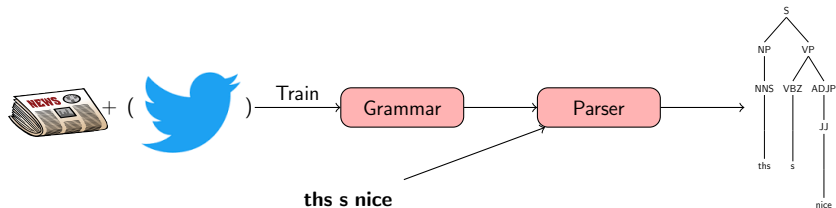
# Problem



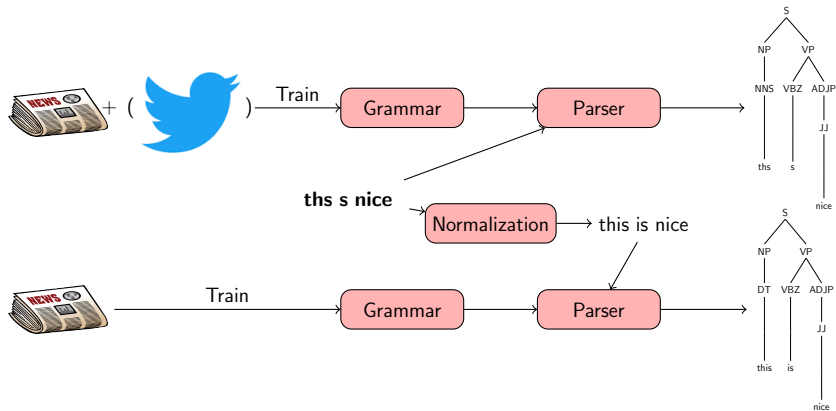
# Idea



# Idea

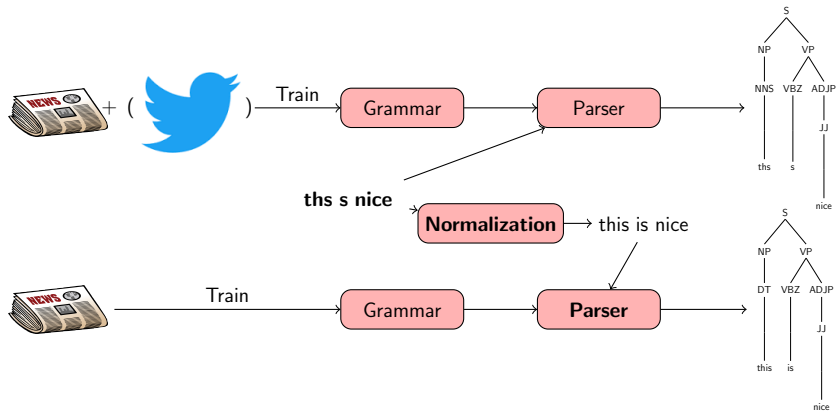


# Idea



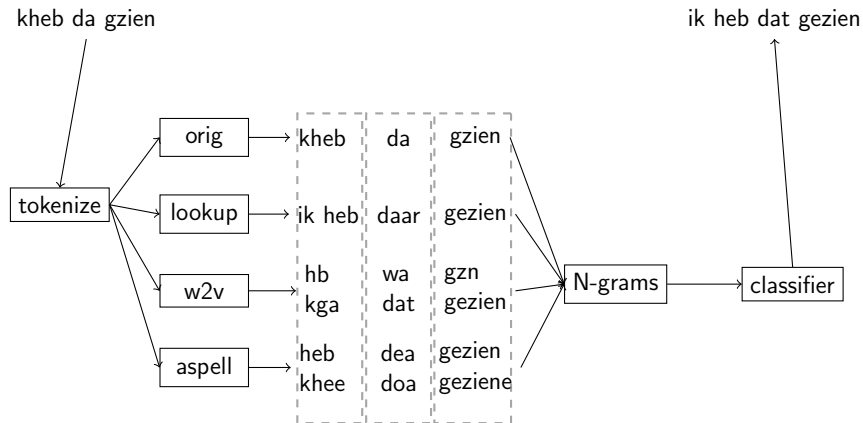


# Idea

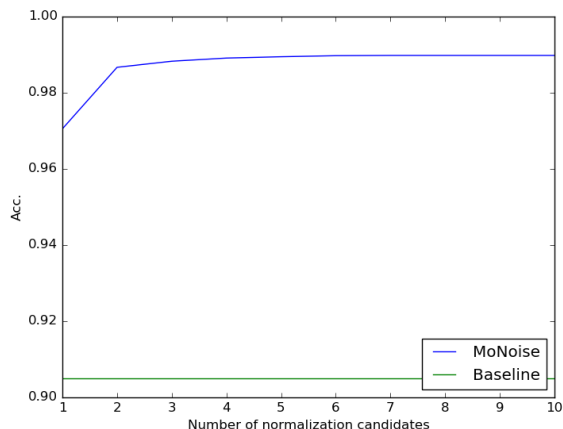


# Normalization

## Mo'Noise



# Normalization



# Normalization

Not found:

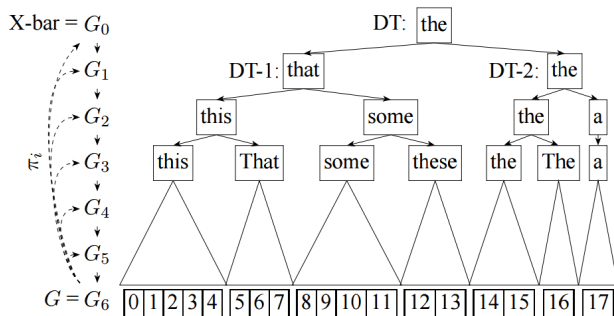
ight	alright
naw	no
acc	actually
shotti	shotgun
ibe	i'm
unliked	disliked
pgh	pittsburgh
1	one

## Dataset:

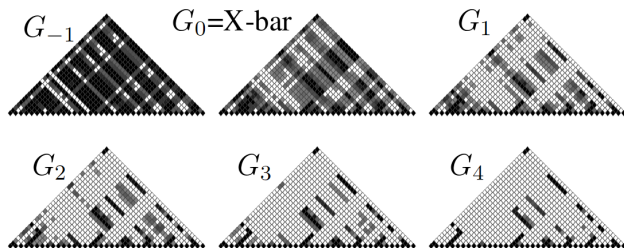
- Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan and Josef van Genabith, 2011. From News to Comment: Resources and Benchmarks for Parsing the Language of Web 2.0.
- 519 tweets (250-269)
- Constituency trees (EWT)
- Less noisy compared to normalization corpora

- Berkeley parser (CYK, PCFG-LA)
- Trained on EWT and WSJ

# Parsing

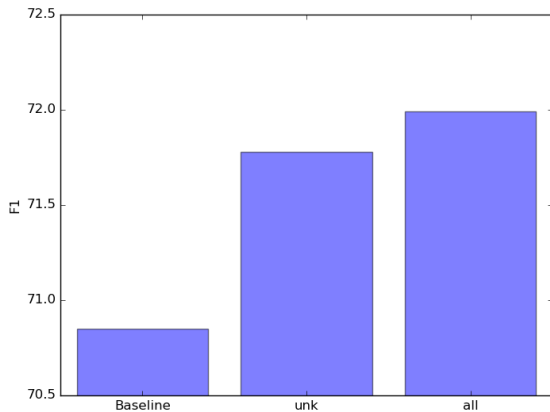


# Parsing





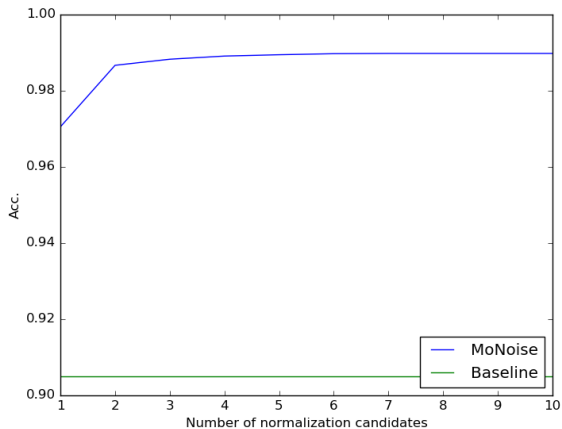
# Parsing



- Nice improvement,
- but:

- Nice improvement,
- but:
- Normalization is not perfect
- Information is lost

# Parsing



# Parsing as Intersection

- Bar-hilel (1961)
- "The intersection of a context-free language with a regular language is again a context-free language"

# Parsing as Intersection

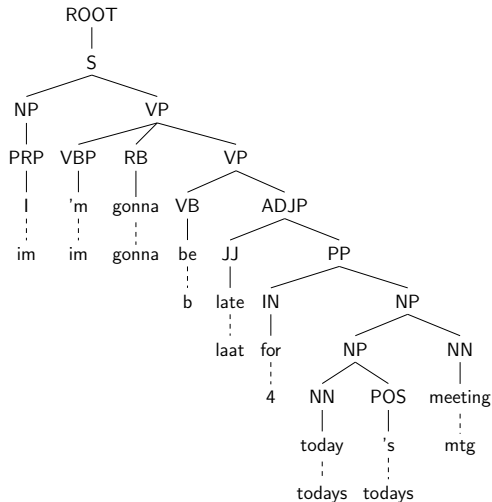
- Bar-hilel (1961)
- "The intersection of a context-free language with a regular language is again a context-free language"
- Ability to find optimal parse tree over a word graph

# Parsing as Intersection

In practice:

- Treat words as constituents

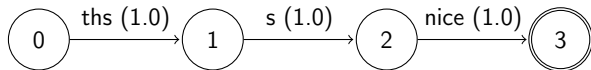
# Parsing as Intersection



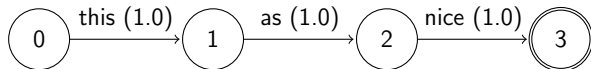


# Parsing as Intersection

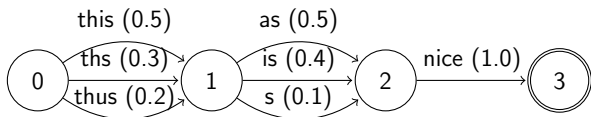
Raw sent:



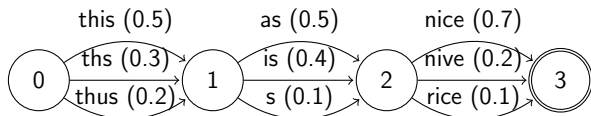
Best norm:



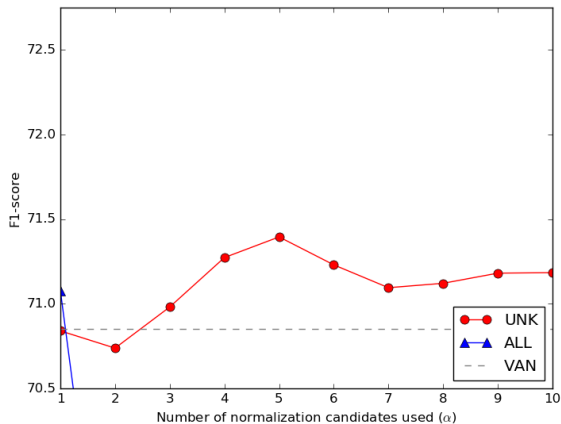
UNK:



ALL:



# Parsing as Intersection



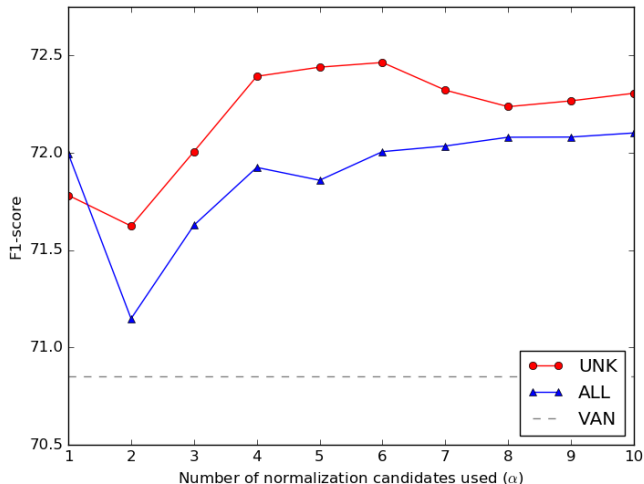
# Parsing as Intersection

Adjust normalization weight:

$$P_{chart} = (1 + \beta^2) * \frac{P_{norm} * P_{pos}}{(\beta^2 * P_{norm}) + P_{pos}} \quad (1)$$

# Evaluation

Development data:



# Evaluation

Test data:

Parser	dev	test
Stanford parser	66.05	61.95
Berkeley parser	70.85	66.52
Best norm. seq.	72.04	66.94
Integrated norm.	72.77	<b>67.36*</b>
Gold POS tags	74.98	71.80

But: normalization does not improve!

- Why?

But: normalization does not improve!

- Why?
- Is this still domain adaptation?

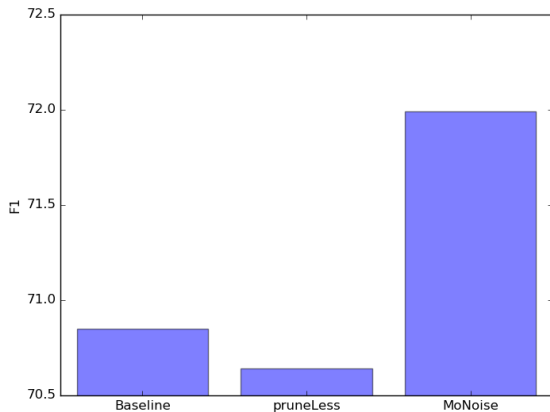
But: normalization does not improve!

- Why?
- Is this still domain adaptation?
- Or do we just prune less?



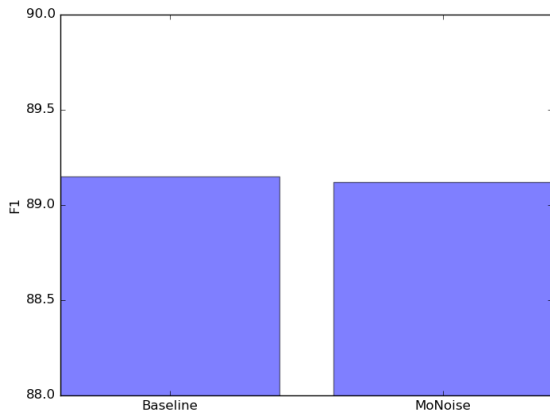
# Evaluation

Does pruning help on this domain?



# Evaluation

Does our model improve parsing of other domains?



# Evaluation

- Why?

# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word

# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word
- Is this still domain adaptation?

# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word
- Is this still domain adaptation?
- ...

# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word
- Is this still domain adaptation?
- ...
- Do we just prune less?

# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word
- Is this still domain adaptation?
- ...
- Do we just prune less?
- Probably not



# Evaluation

- Why?
- Sometimes, we use wrong normalizations that share syntactic properties with the original word
- Is this still domain adaptation?
- ...
- Do we just prune less?
- Probably not
- Don't forget: the normalization is already quite good!

# Evaluation

<https://bitbucket.org/robvandergerg/monoise>

# Conclusion

- Word embeddings and aspell complement each other well for the normalization task
- A random forest classifier works very well for ranking normalization candidates
- Normalization most useful when integrated into the parser
- However, the improvement is not always a result of the correct normalization
- If integration is not an option; do not filter the words before normalization

# Conclusion

Future work:

- Improve normalization by parsing
- Unsupervised normalization
- Reranking (lexicalized parsing?)
- How can we adapt RNN-parsers

# Conclusion

