Parser Adaptation for Social Media by Integrating Normalization

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

03-03-2017
Hey Gj, im gonna b laat 4 todays mtg
Hey Gj, im gonna b laat 4 todays mtg

aight m8
Problem
Problem
I’m gonna be late for today’s meeting today’s mtg today’s todays todays
Idea

+ ( ) → Train → Grammar → Parser

ths s nice

Rob van der Goot
r.van.der.goot@rug.nl
03-03-2017
Idea

The diagram illustrates the process of training a grammar and parser system. The input is a Twitter message, represented as "ths s nice". The system first goes through a normalization step to convert the message to "this is nice". This normalized message is then fed into a grammar and parser system. The output of the parser is a syntactic tree, shown on the right side of the diagram. The tree structure breaks down the sentence into its constituent parts: noun phrase (NP), verb phrase (VP), adjective phrase (ADJP), and the verb (VBZ).
Idea

This is nice

Normalization

this is nice
Normalization

Mo’Noise

tokenize

orig

lookup

w2v

aspell

N-grams

classifier

kheb da gezien

ik heb dat gezien

Rob van der Goot
r.van.der.goot@rug.nl
Normalization

![Graph showing the accuracy of MoNoise and Baseline vs the number of normalization candidates. The accuracy for MoNoise starts at a lower value than the baseline and increases as the number of candidates increases, eventually converging to a higher value than the baseline.](image-url)
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

\[ X\text{-}bar = G_0 \]

\[ \pi_i \rightarrow G_3 \rightarrow G_4 \rightarrow G_5 \rightarrow G = G_6 \]

DT: the

DT-1: that

\[ \text{this} \rightarrow \text{That} \rightarrow \text{some} \rightarrow \text{these} \]

DT-2: the

\[ \text{the} \rightarrow \text{The} \rightarrow \text{a} \]
Parsing

$G_{-1}$  $G_0=X$-bar  $G_1$

$G_2$  $G_3$  $G_4$
Parsing

![Bar chart showing the F1 scores for Baseline, unk, and all categories.](image-url)
Nice improvement,
but:
Nice improvement,
but:
Normalization is not perfect
Information is lost
Parsing

![Chart showing accuracy over number of normalization candidates for MoNoise and Baseline approaches.](chart.png)
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
In practice:
- Treat words as constituents
I'm gonna be late for tomorrow's meeting.
Parsing as Intersection

Raw sent:

0 -> ths (1.0) -> 1 -> s (1.0) -> 2 -> nice (1.0) -> 3

Best norm:

0 -> this (1.0) -> 1 -> as (1.0) -> 2 -> nice (1.0) -> 3

UNK:

0 -> ths (0.3) -> this (0.5) -> 1 -> thus (0.2) -> 1 -> as (0.5) -> 2 -> is (0.4) -> 3

ALL:

0 -> ths (0.3) -> this (0.5) -> thus (0.2) -> 1 -> s (0.1) -> 2 -> nice (0.7) -> 3 -> rice (0.1)
Parsing as Intersection

![Graph showing F1 score against number of normalization candidates used (α).](image-url)
Adjust normalization weight:

\[ P_{chart} = (1 + \beta^2) \cdot \frac{P_{norm} \cdot P_{pos}}{(\beta^2 \cdot P_{norm}) + P_{pos}} \]  

(1)
Evaluation

Development data:

![Graph showing F1-score vs. number of normalization candidates used](image)

- UNK
- ALL
- VAN
Test data:

<table>
<thead>
<tr>
<th>Parser</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford parser</td>
<td>66.05</td>
<td>61.95</td>
</tr>
<tr>
<td>Berkeley parser</td>
<td>70.85</td>
<td>66.52</td>
</tr>
<tr>
<td>Best norm. seq.</td>
<td>72.04</td>
<td>66.94</td>
</tr>
<tr>
<td>Integrated norm.</td>
<td>72.77</td>
<td>67.36*</td>
</tr>
<tr>
<td>Gold POS tags</td>
<td>74.98</td>
<td>71.80</td>
</tr>
</tbody>
</table>
Evaluation

But: normalization does not improve!

Why?
Evaluation

But: normalization does not improve!

- Why?
- Is this still domain adaptation?
Evaluation

But: normalization does not improve!

- Why?
- Is this still domain adaptation?
- Or do we just prune less?
Does pruning help on this domain?

![Bar chart showing comparison between Baseline, pruneLess, and MoNoise.]
Evaluation

Does our model improve parsing of other domains?

![Bar chart comparing Baseline vs MoNoise performance](chart.png)
Why?
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?
- ...
Why?

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

Is this still domain adaptation?

... 

Do we just prune less?
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/robvanderg/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

thx for ur attention.
#anyquestions?