Extracting User-Reported Mobile Application Defects from Online Reviews

Yue Wang
University of Delaware
U.S.A

Hongning Wang
University of Virginia
U.S.A

Hui Fang
University of Delaware
U.S.A

InfoLab

University of Delaware

Defects identification is important for developers

Developers

Verbose reviews are hard to digest in a short time

Fun but too many ads ★★★★
by Azalard

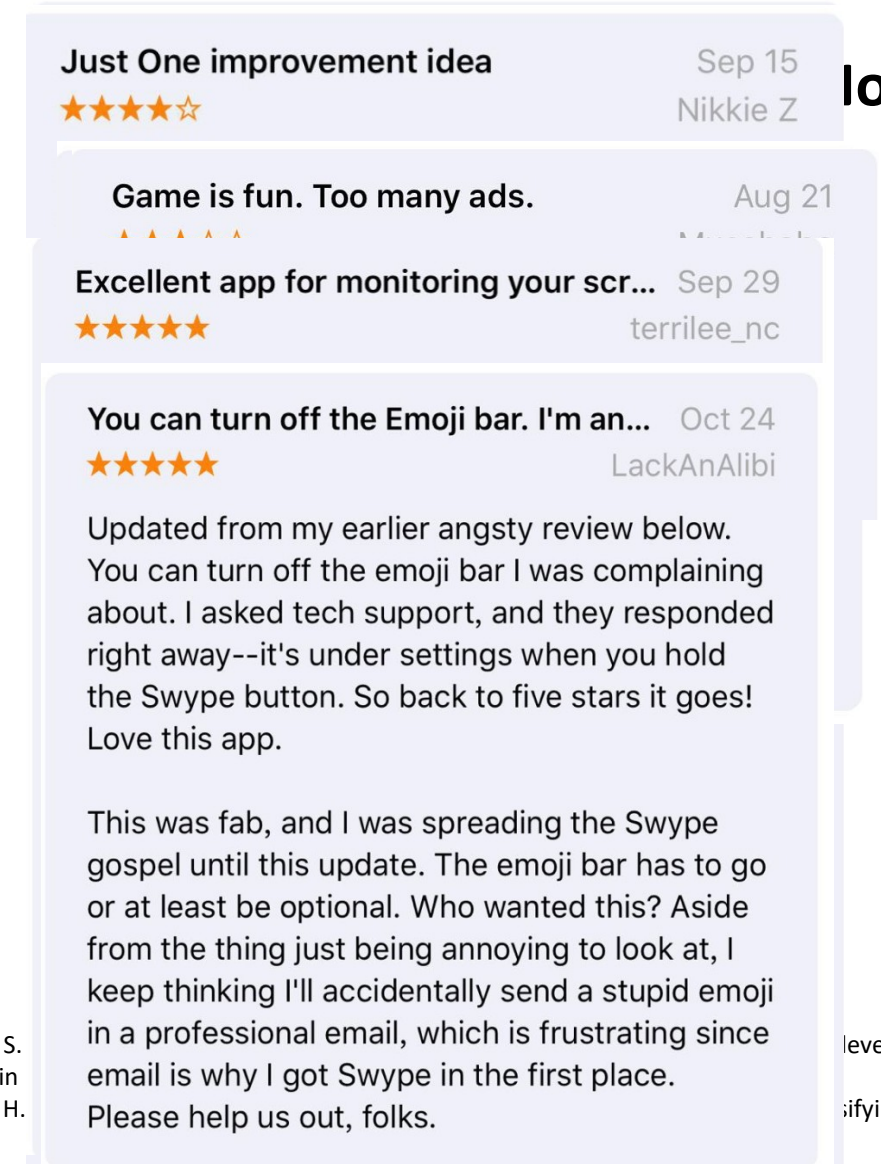
This is a fantastic game for both children and adults. It's challenging, entertaining, and fairly easy to play. And unlike other games, you don't have to spend real money in order to purchase cheats or power-ups. You can progress on your own skill. It's an enjoyable way to pass the time on a road trip, a lazy afternoon, or when you just need a break and want something mindless to do.

The game does have a few issues. ***For me, it crashes quite frequently. It's a regular occurrence.*** The game also has these annoying ads and videos that pop-up at random. I realize that these ads are what gives us all this free content, but they often appear right when you're trying to move from puzzle to puzzle. It's like a kid running right out in the road while you're driving down the street. It's unpredictable and unexpected. What's worse is that the ads will still play sound even when you turn off all the music and sound in the app. [illegible] to them when I have my headset plugged in is a bit much.

Could we identify those defects automatically?

Existing solutions[1][2] focus on review level classification

Defects reports



Ion defects reports

[1] N. Chen, J. Lin, S. ... levelopers from mobile app marketplace," in

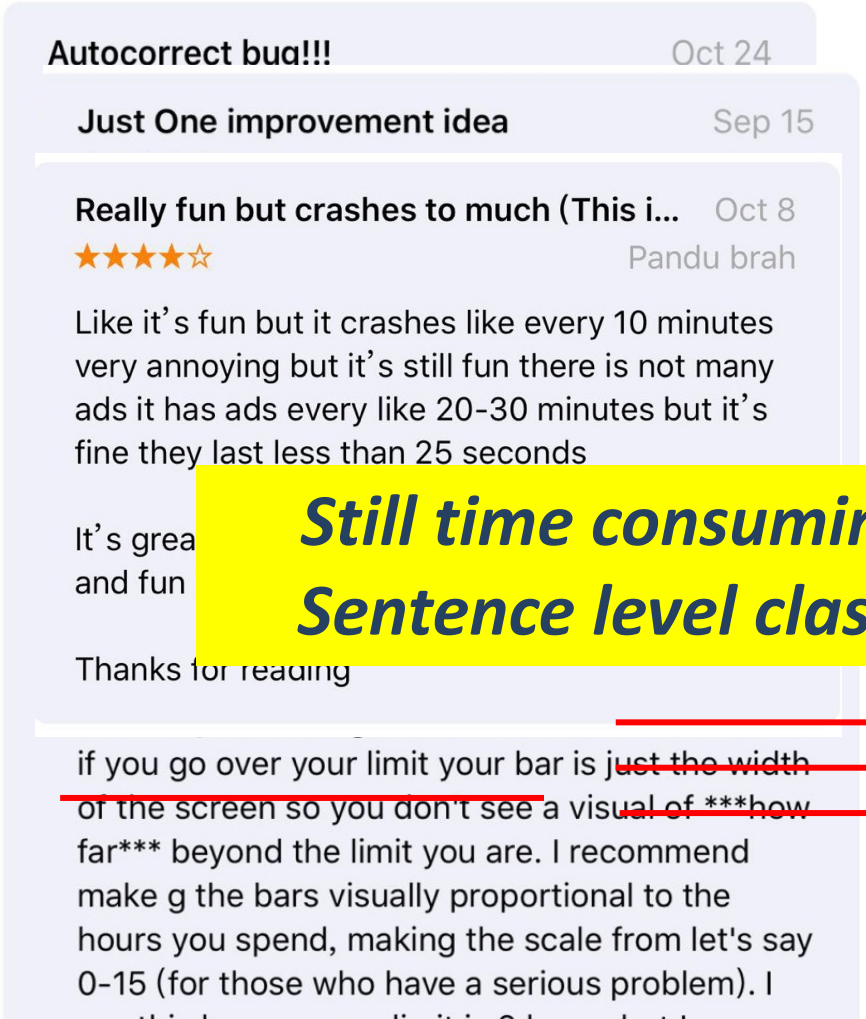
[2] W. Maalej and H. ... sifying app reviews," in IEEE RE 2015, 2015.

InfoLab

UNIVERSITY OF DELAWARE

Existing solutions focus on review level classification

Defects reports



Still time consuming and labor intensive!
Sentence level classification is necessary!

Contribution

- We developed a structured learning solution based on hidden structural support vector machines.
- The trained model utilize the dependency between sentences and reviews to predict the defects status at sentence level.
- The proposed method could be trained with partially annotated data.

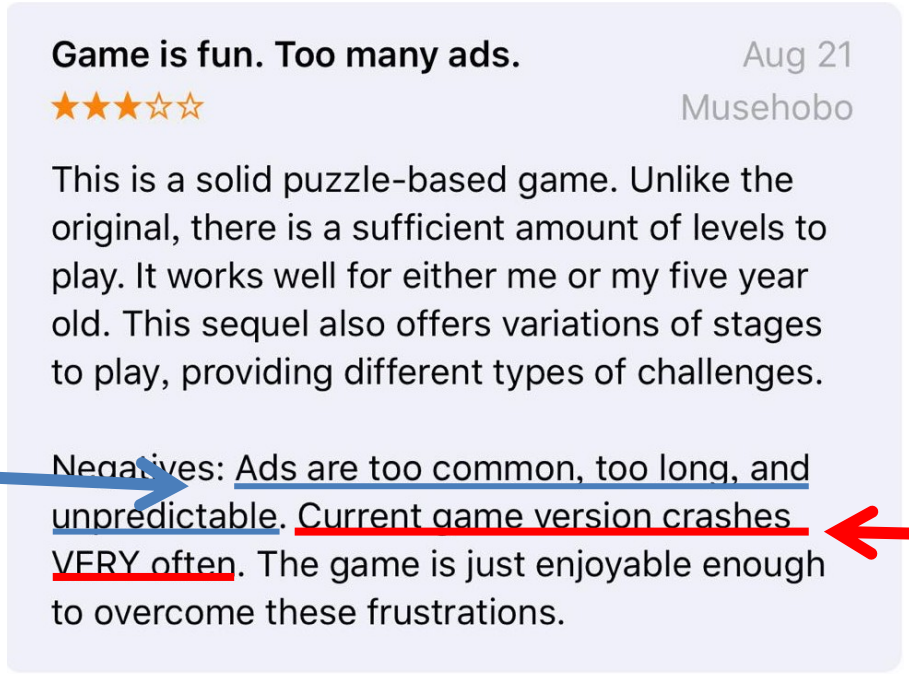
Problem formulation

- Sentence level defects reporting classification
  - Given a user-generated review $R = \{s_1; s_2; \ldots; s_n\}$, consisting of n sentences, the sentence level defects classification generate a binary label for each sentence $s_i = 1$, indicating this sentence is reporting a defect, otherwise $s_i = 0$.

S1)All emails from my director won't open. S2)Gives me an error message. S3)Also, it would be nice if you could have a setting for what language alerts you receive. S4)Although it is nice to have 10 new alerts on my phone all at once, I don't need the same one in 3 languages. S5)And since the newest update, I can't see anything under the my business tab. S6)It's just white.

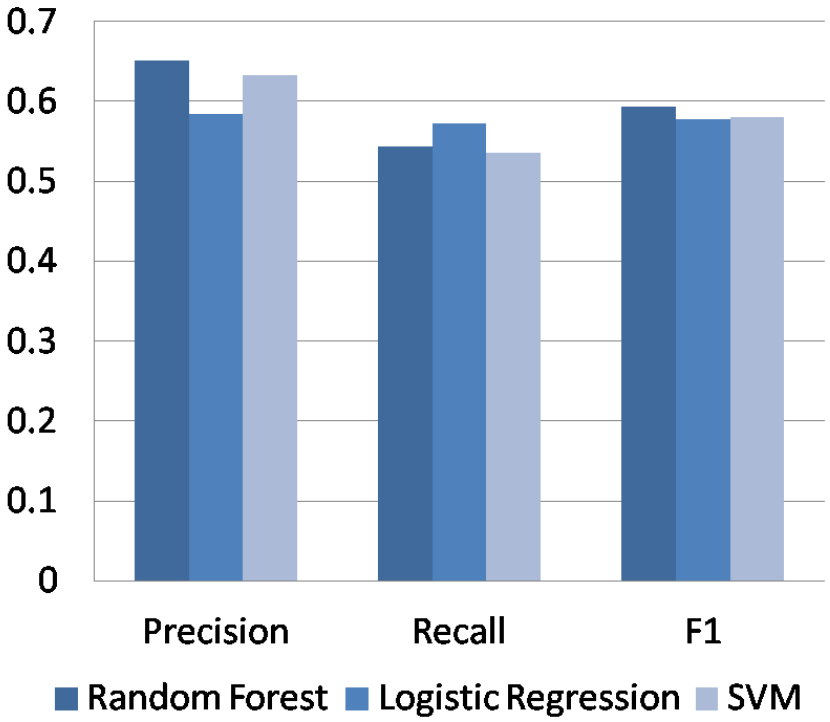
Problem formulation

- Defects:
  - The abnormal behavior of an app which produces an incorrect or unexpected result, or behaves in unintended ways.



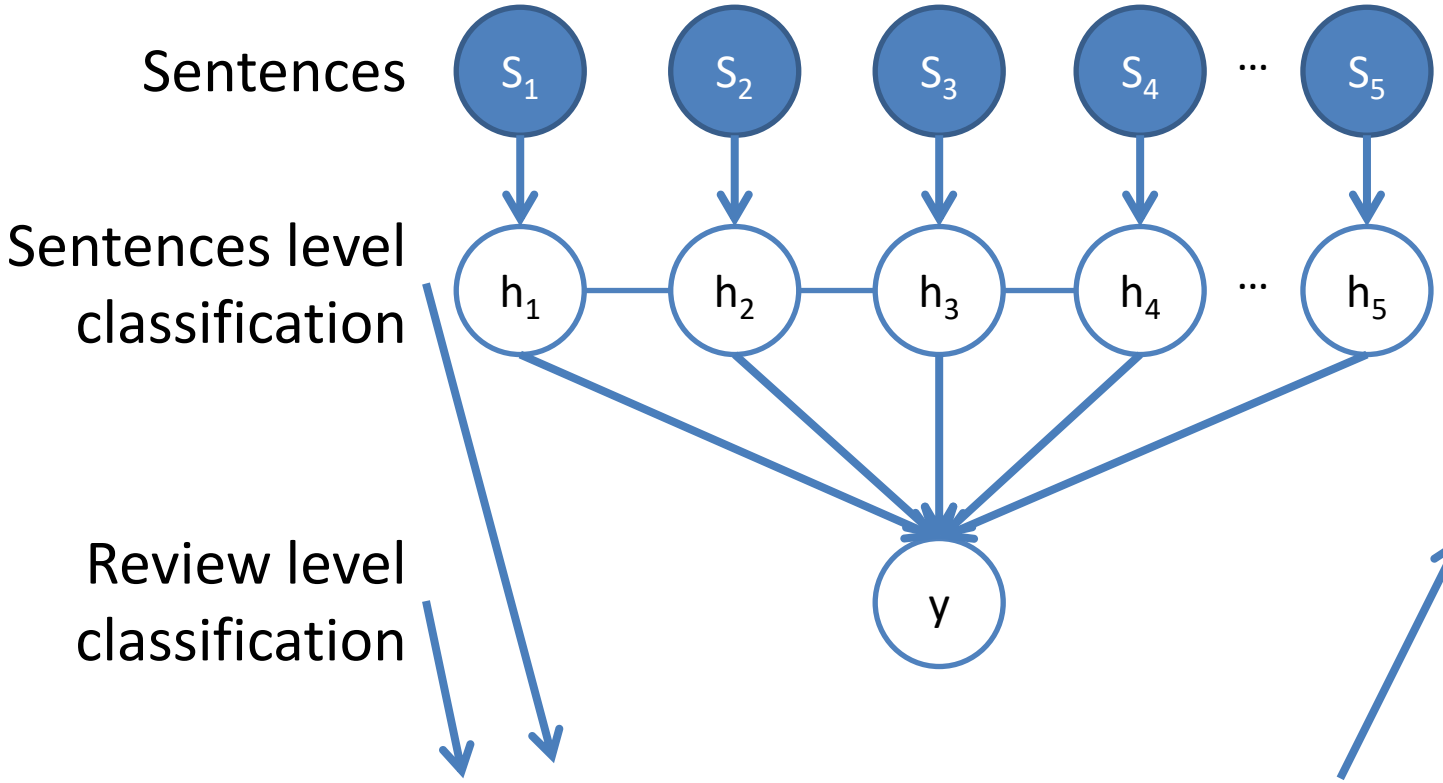
A straightforward solution

- Review level classifier on sentence level classification



Unsatisfying performance due to the lack of connections between sentence and reviews

Sentence level defects classification with latent structures (hSVM)



Sentences level features: *Aspects, Length, Sentiment, etc*

Review level features: *Rating, Similarity, etc*

$$(\hat{y}, \hat{H}) = \underset{(y,H)\in\mathcal{Y}\times\mathcal{H}}{\arg\max} \; \omega^{\top}\Phi(R, H, y)$$

The constraints between sentences and review

$$\text{if } \hat{y} = 1 \quad \Rightarrow \quad \sum_{\hat{h} \in \hat{H}} \neq 0$$

Defect reporting review must contain at least one defect reporting sentence

$$\text{if } \hat{y} = 0 \quad \Rightarrow \quad \sum_{\hat{h} \in \hat{H}} = 0$$

Non defect reporting review cannot contain any defect reporting sentence

$$\sum_{i=1}^{n} \hat{h}_i \geq \hat{y}$$

Model training

- Estimate w with structured SVM framework[3]

$$\min_{\omega,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{m=1}^{M}\xi_m$$

$$s.t. \quad \forall m, \max_{H\in\mathscr{H}} \omega^\top \Phi(R_m, H, y_m) \geq$$

$$\max_{(\hat{y},\hat{H})\in\mathscr{Y}\times\mathscr{H}} \omega^\top \Phi(R_m, \hat{H}, \hat{y}) + \Delta(y_m, \hat{y}, H_m, \hat{H}) - \xi_m$$

$$\xi_m \geq 0$$

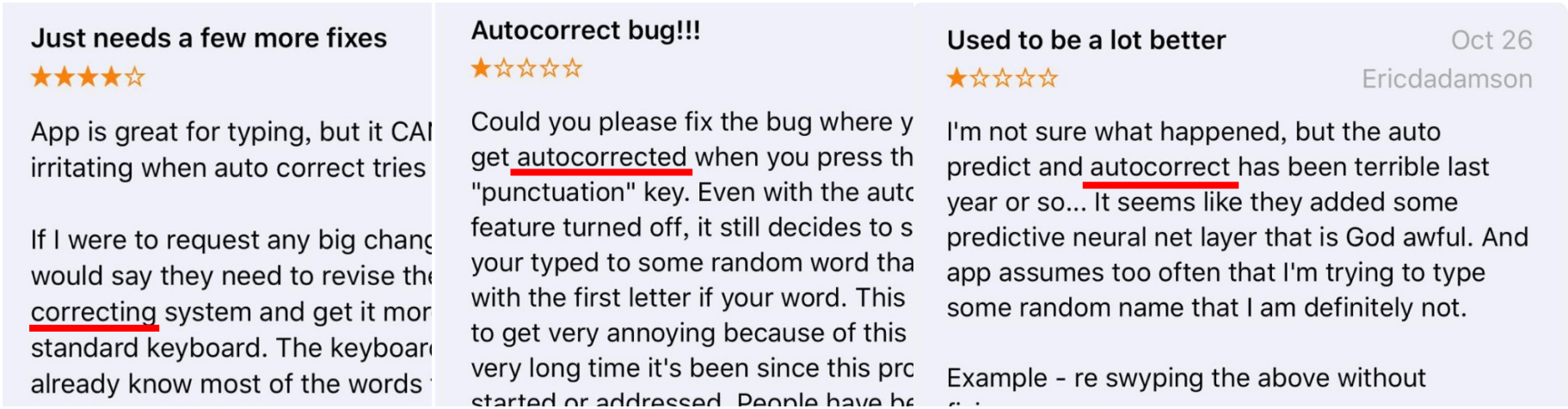
Measure error between prediction and ground-truth

$$\sum_{i=1}^{n} \hat{h}_i \geq \hat{y}$$

[3] M.-W. Chang, V. Srikumar, D. Goldwasser, and D. Roth, "Structured output learning with indirect supervision," in Proceedings of ICML, 2010.

Sentence level features

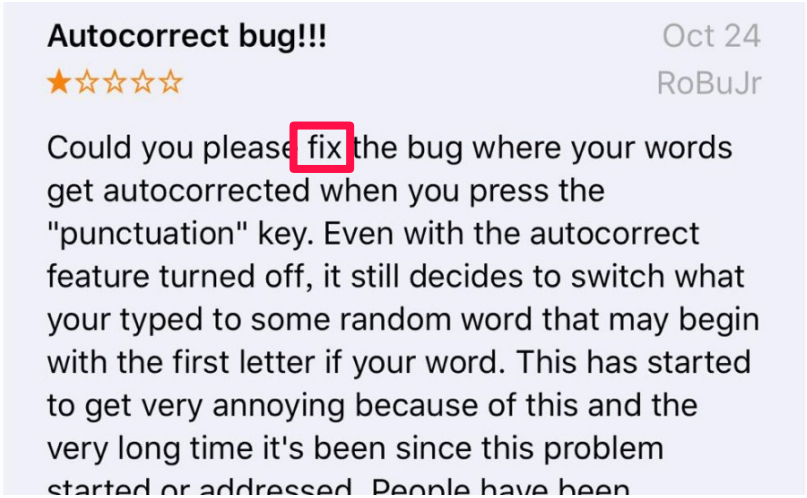
- Aspects
  - User usually mention one or more specific aspects of the app if it is defective.

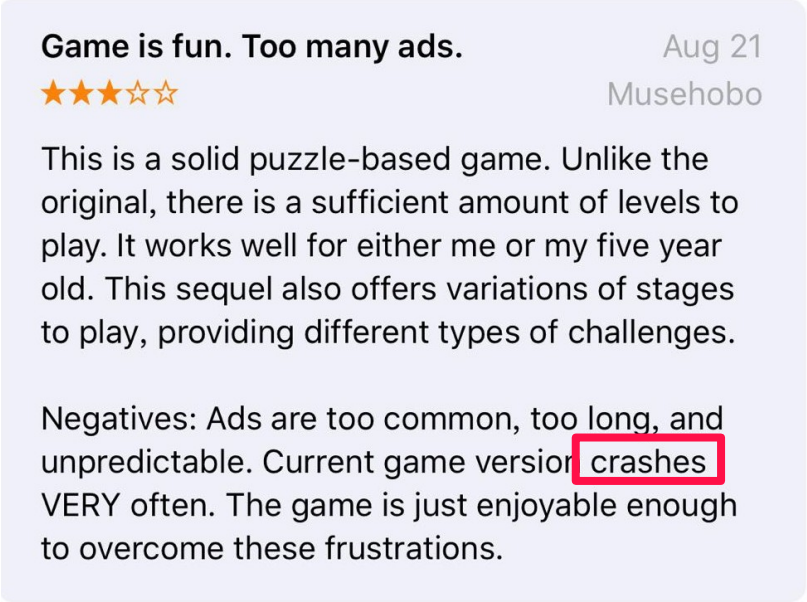


Trained PLSA topic model to recognize the aspects from the reviews

Sentence level features

- Indicator words
  - A list of words that shared by the users when they tend to report the defects.
  - *Freeze, crush, idle, etc.*

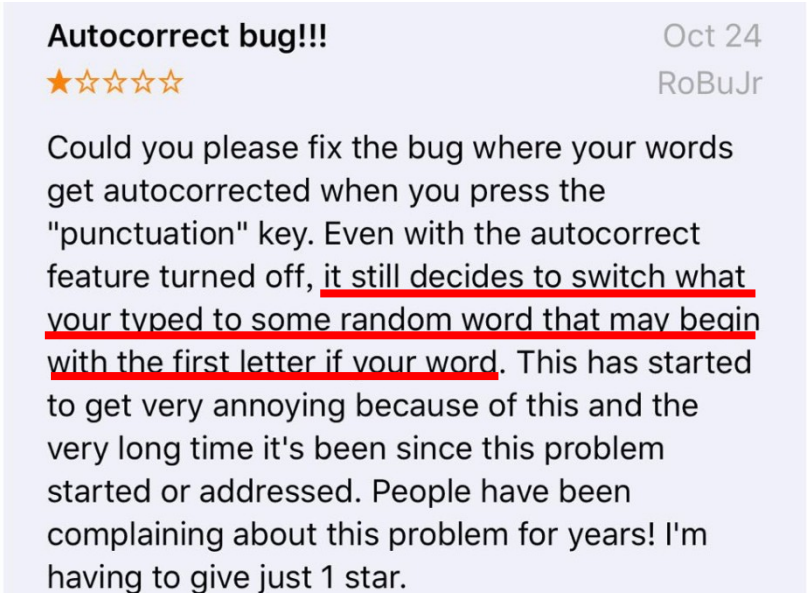


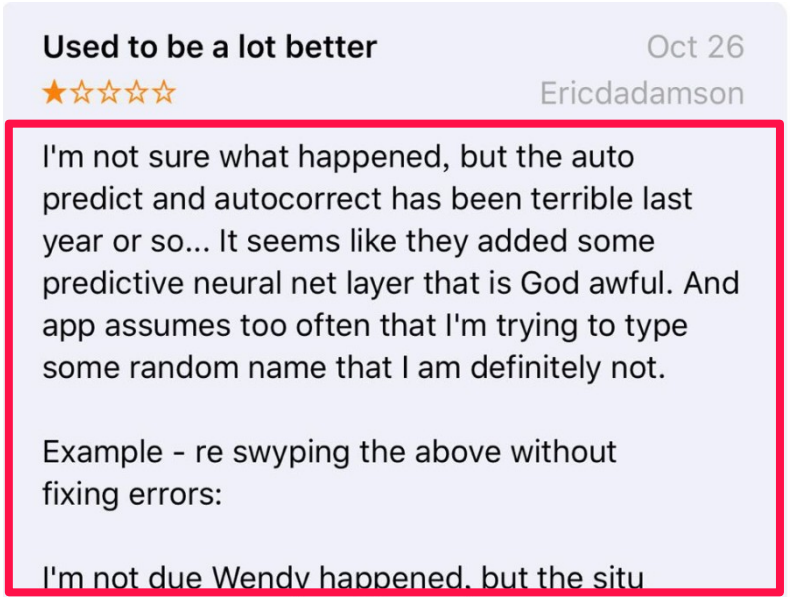


Created an indicator word list based on the training data

Sentence level features

- Similarity with sentences in the other reviews
  - Users would report the same defects in a similar way

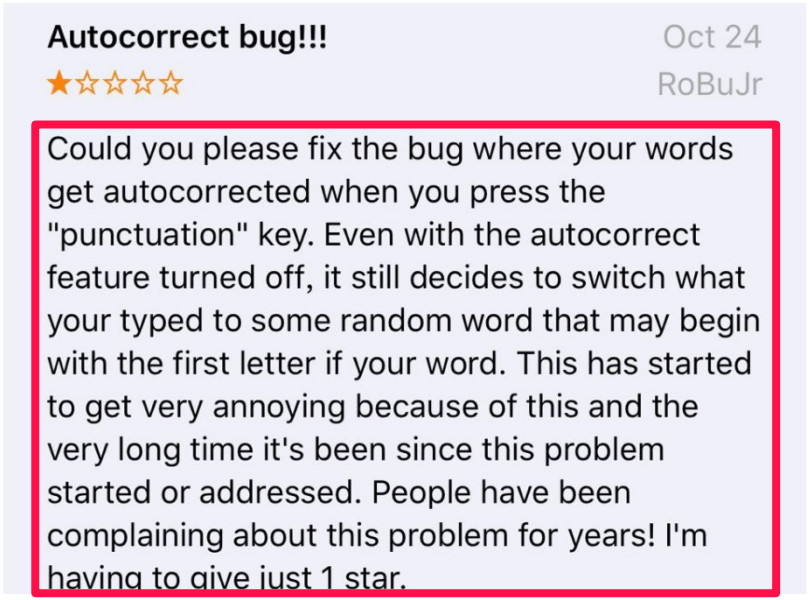


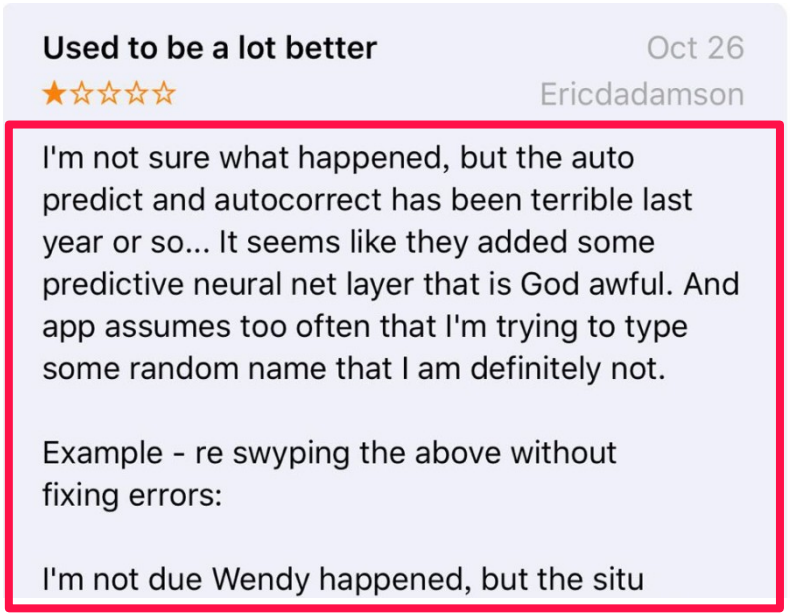


Highest sentence to review similarity is used as feature

Review level features

- Similarity with the other reviews





Highest review to review similarity is used as feature

All features

- 14 sentence level features
  - Contains negation words?
  - Overall sentiment orientation
  - Sentence length
  - etc…
- 8 review level features
  - Aspect coverage
  - Aspect consistence
  - Indicator consistence
  - etc…

Data sets

- Apps crawled from Apple App Store
  - Fields being crawled: App name, description, reviews, etc
  - Fully labeled reviews and Partially labeled reviews

	Fully labeled reviews	Partially labeled reviews
Number of **reviews** (defect / non-defect)	274/572	400/818
Number of **sentences** (defect / non-defect)	557/3917	84/441
Unlabeled sentences	0	8968

Performance on Fully annotated reviews

- hSVM model outperformed most of the baselines

	Prec	Recall	F1
J48-S[4]	0.716	0.863	0.783
LR-S	0.799	0.811	0.805
SVM-S	0.785	0.832	0.808
RF-S	0.882	**0.947**	**0.913**
hSVM	**0.919**	0.898	0.908

Although Random Forest outperforms hSVM on recall, it requires much more training data

[4] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, ""how can i improve my app? classifying user reviews for software maintenance and evolution",," in ICSME '15, 2015.

Performance on Fully annotated reviews

- hSVM model outperformed all the baselines on ***short*** sentences

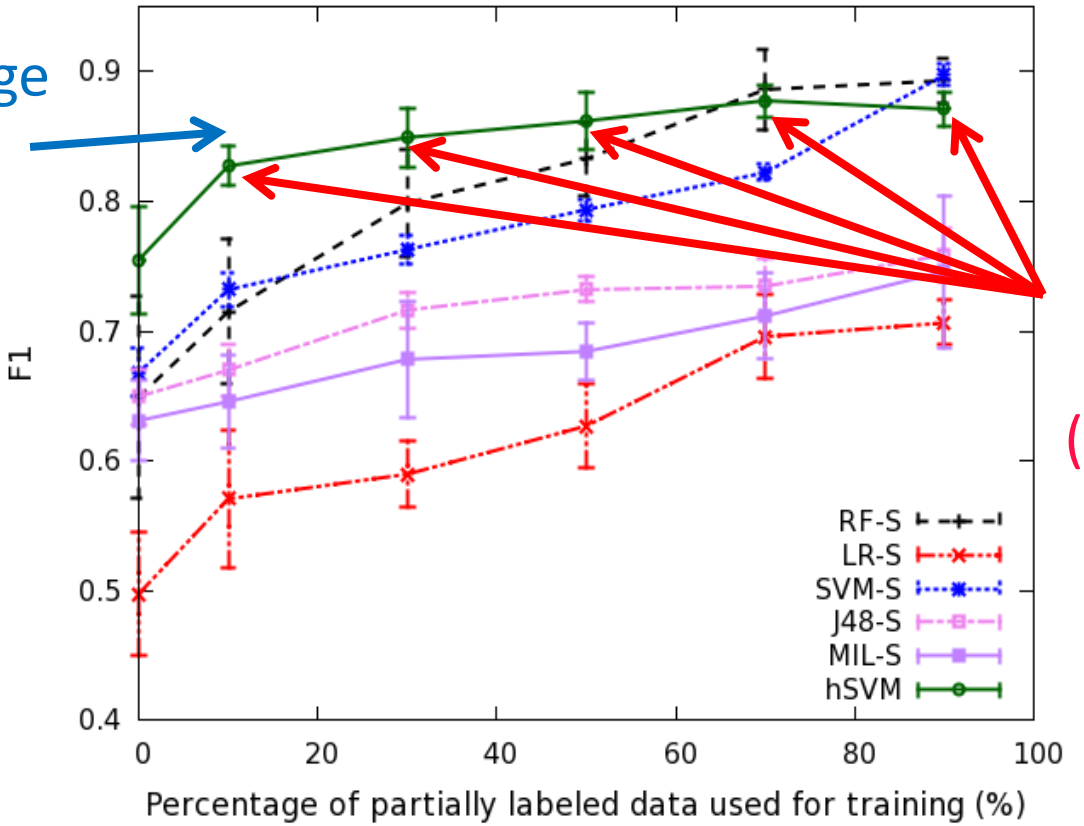
	Prec	Recall	F1
J48-S	0.672	0.826	0.741
LR-S	0.761	0.792	0.776
SVM-S	0.763	0.814	0.781
RF-S	0.854	0.893	0.873
hSVM	**0.912**	**0.931**	**0.921**

The improvement of our method comes from utilizing the dependency betweens sentences and review to predict the sentence level labels.

Performance on Partially annotated data

Quickly converge to optimal performance

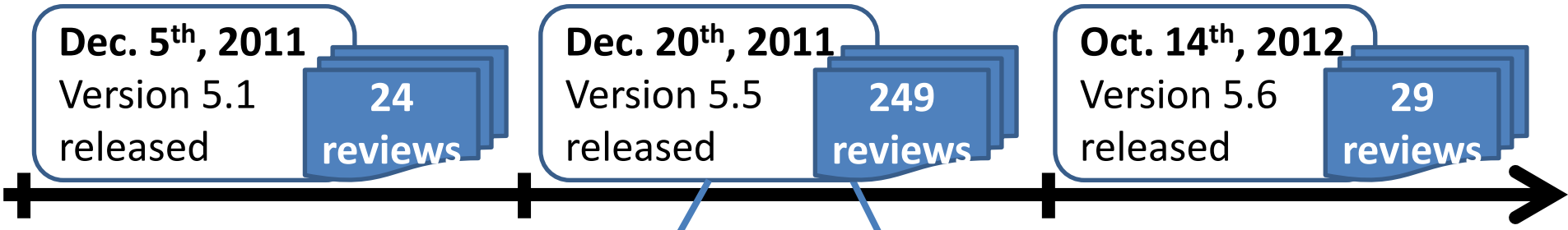
(Require less training data)



Smaller variance

(More robust)

hSVM model could learn from the partially annotated data and less likely to overfit to a particular training set



Description of version 5.5

The Microsoft Tag app lets you instantly connect to a whole new world of information and entertainment. ***This latest version now reads Microsoft Tag barcodes and QR Codes****. No need to type long URLs or text short codes;* ***simply scan a 2D barcode wherever you see one****. Tag recognition technology transforms traditional marketing,*

…

as well as view content in an embedded browser and share items via email, Facebook and Twitter. ***Plus, it allows you to save and edit contact information.***

Identified defect reporting sentences

69 sentences mentioning "QR code":

Close when read QR code.

Can't read the code because it closes every time!

137 sentences mentioning "scan":

This crashes before I can scan, it has a bug please fix.

It crashes after I tapped the scan button

2 sentences mentions "save":

I can not save the information from the link!

The save and edit is a crap!

Conclusion and Future Work

- Conclusion
  - We proposed a structured learning solution to address the problem of identifying sentence-level defects reports.
  - The proposed method utilize the dependency between sentences and reviews to better predict the sentence level label
- Future work
  - Study how to include unlabeled data for training
  - Summarize the extracted sentences and map to aspects
  - Investigate the usage of update history

Thank you!
Q&A

Corresponding Author
Yue Wang (wangyue@udel.edu)