A time-series classification model for Twitter opinion lexicon expansion

Felipe Bravo Márquez
Chief Supervisor: Associate Professor Bernhard Pfahringer
Supervisor: Associate Professor Eibe Frank

University of Waikato
Computer Science Department

November 7, 2014

Social Media

- Microblogging services are increasingly being adopted by people in order to access and publish information.
- **Twitter**: Massively used Microblogging platform where users post messages limited to 140 characters.
- Twitter users tend to publish **personal opinions** regarding certain topics and news events.

Opinion Mining or Sentiment Analysis

- Application of **NLP** and **text mining** techniques to identify and extract subjective information from textual datasets.

Sentiment Classification Problem

1. Automatically classify a textual message to classes **positive**, **negative**, or **neutral**.

Approaches

- Most methods rely on opinion lexicons.
- An opinion lexicon is a lists of terms labelled by sentiment.
- They are normally composed of positive and negative words such as **happy** and **sad**.

Sentiment Analysis and Social Media

- Opinions are provided **freely and voluntarily** by the users in Twitter.
- Analysing the sentiment underlying these opinions has important applications in product marketing and politics.
- The words used in Twitter include many abbreviations, acronyms, and misspelled words.
- This words are **not** covered by most popular lexicons.
- The manual creation of a Twitter-oriented opinion lexicon is a **time-consuming** task.

Proposal

- We propose a **supervised framework** for opinion lexicon expansion for **Twitter**.
- Each expanded word has a **probability distribution**, describing how positive, negative, and neutral it is.
- All the entries of the lexicon are associated with a corresponding **part-of-speech** tag.
- This is useful for word disambiguation e.g., apple can be a company or a fruit.
- This is the first lexical resource for Twitter with these properties.
- These properties are inspired by **SentiWordnet**.

Methodology

1. **Collect** tweets from the domain and the time period for which the lexicon needs to be expanded.
2. Label the collection with sentiment classes in an **automatic** way.
3. **Tag** all the words using a part-of-speech tagger.
4. Calculate word-level **time-series** for all tagged words and extract **sentiment features** from them.
5. Label the sentiment of the words that match an **existing hand-made** polarity lexicon.
6. Train a **word-level classifier** using the word-level features and the words labels from the seed lexicon.
7. Use the trained classifier to **estimate** the polarity distribution of the remaining unlabelled words.

Ground-Truth word polarities

- The expansion requires a **seed lexicon** with words labelled by sentiment.
- We create a meta-lexicon by taking the **union** of existing hand-made lexicons.
- We discard all words where a **polarity clash** is observed.

	Positive	Negative	Neutral
AFINN	564	964	0
Bing Liu	2003	4782	0
MPQA	2295	4148	424
NRC-Emo	2312	3324	7714
Union	4331	7004	8013
Meta-Lex	3730	6368	7088

Table : Lexicon Statistics

Ground-Truth word polarities (2)

Figure : Polarity clashes

Obtaining labelled tweets

- We **require** a collection of time-stamped tweets with their corresponding **polarity labels**.
- Tweets can be collected from the Twitter API.
- Tweets exhibiting **positive :)** and **negative :(** emoticons are labelled according to the emoticon's polarity.
- We consider **two** collections of tweets covering multiple topics: The **Edinburgh corpus** (ED), and the **Stanford Sentiment corpus** (STS).

	ED	STS
Positive	1,813,705	800,000
Negative	324,917	800,000
Total	2,138,622	1,600,000

Table : Collection statistics

Word-level Time-Series

- To train the word-level classifier we need to **calculate features** from each word found in the collection of tweets.
- Our features exploit the **temporal structure** of the collection of labelled tweets.
- Tweets are lowercased, tokenised and POS-tagged.
- We prepend a **POS-tag** prefix to each word in order to differentiate **homographs** exhibiting different POS-tags.
- We create two types of **time-series** for each word: the **Stochastic Gradient Descent** (SGD) series, and the **Semantic Orientation** (SO) series.

The SGD time-series

- This time-series is calculated by incrementally training a **linear support vector machine** from the collection of labelled tweets.
- We use **stochastic gradient descent** (SGD) online learning process.

$$\frac{\lambda}{2}||w||^2 + \sum[1 - y(\mathbf{xw} + b)]_+. \quad (1)$$

- The weights of this linear model correspond to POS-tagged words and are updated in an **incremental fashion**.
- The model's weights determine how strongly the absence or presence of a word **influences** the prediction of **polarity** classes.
- We use **time windows** of $1,000$ examples.

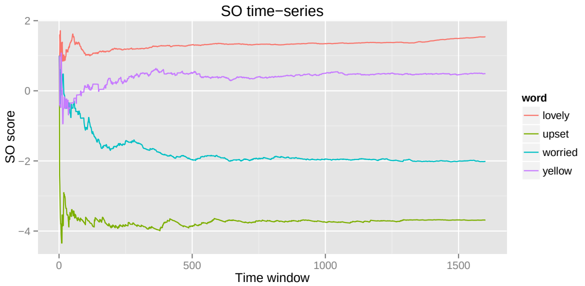
The SO time-series

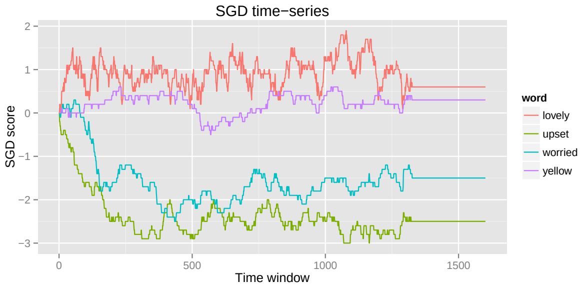
- The second time-series corresponds to the **accumulated semantic orientation** (SO).
- It is based on the **point-wise mutual information** measure.

$$\mathrm{SO}(\mathit{word}) = \mathit{log}_2 \left(\frac{\mathrm{count}(\mathrm{word} \wedge \mathrm{pos}) \times \mathrm{count}(\mathrm{neg})}{\mathrm{count}(\mathrm{word} \wedge \mathrm{neg}) \times \mathrm{count}(\mathrm{pos})} \right) \tag{2}$$

- We use time windows of $1,000$ examples and the **Laplace** correction to avoid the zero-frequency problem.

Word-level Time-Series





Word-level Features

- We extract word-level **attributes** from both SGD and SO time-series.

Feature	Description
mean	The mean of the time-series.
trunc.mean	The truncated mean of the time-series.
median	The median of the time-series
last.element	The last observation of the time-series.
sd	The standard deviation of the time-series .
iqr	The inter-quartile range.
sg	The fraction of times the time-series changes its sign.
sg.diff	The sg value for the differenced time-series.

Table : Time-series features

- We also include the POS-tag of the word as a nominal attribute.
- To create training data for machine learning, all the words **matching** the metalexicon are **labelled** according to the lexicon's polarities.

Training data example

Attribute	A-lovely	A-yellow	A-upset	V-worried
sgd.last	0.6	0.3	-2.5	-1.5
sgd.mean	0.9	0.2	-2.4	-1.6
sgd.trimm.mean	0.9	0.2	-2.5	-1.6
sgd.median	0.9	0.3	-2.5	-1.6
sgd.sd	0.3	0.2	0.5	0.5
sgd.sg	0.0	0.0	0.0	0.0
sgd.sg.diff	0.2	0.0	0.1	0.0
sgd.iqr	0.5	0.3	0.3	0.3
so.last	1.5	0.5	-3.7	-2.0
so.mean	1.3	0.4	-3.7	-1.8
so.trimm.mean	1.3	0.4	-3.7	-1.9
so.median	1.3	0.5	-3.7	-1.9
so.sd	0.1	0.2	0.2	0.4
so.sg	0.0	0.0	0.0	0.0
so.sg.diff	0.5	0.4	0.4	0.4
so.iqr	0.1	0.1	0.1	0.1
pos.tag	adjective	adjective	adjective	verb
label	positive	neutral	negative	negative

Table : Word-level feature example.

Feature Visualisation

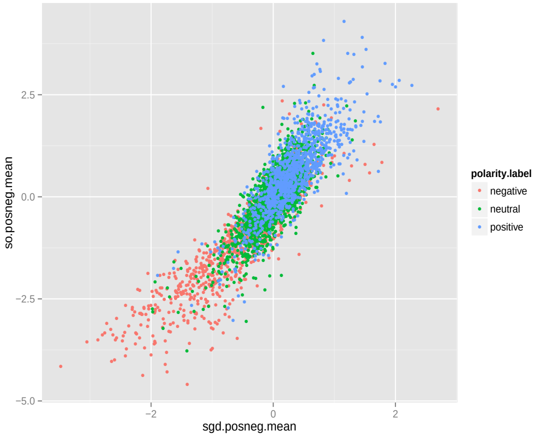
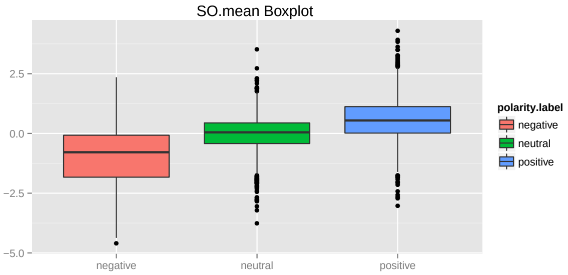
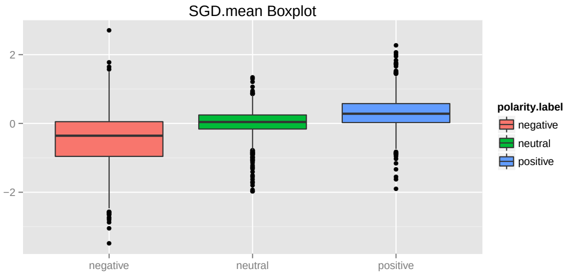


Figure : SO vs SGD scatterplot.

Feature Visualisation (2)





Word-level classification

- We study **three word-level** classification problems.
- *Neutrality*: Classify words as neutral (objective) or non-neutral (subjective).
- *PosNeg*: Classify words to positive or negative classes.
- *Polarity*: Classify words to classes positive, negative or neutral. This is the classification problem we aim to solve.
- We trained **RBF SVM classifiers** for the different problems in both datasets.

Word-level classification (2)

Accuracy					
Dataset	SO	ALL	SGD.TS+POS	SO.TS+POS	SO+POS
ED-Neutrality	61.52 ± 2.21	**65.16 ± 2.09** ∘	64.55 ± 2.27 ∘	64.9 ± 2.14 ∘	64.18 ± 2.12 ∘
ED-PosNeg	74.78 ± 2.93	**76.04 ± 2.72**	73.61 ± 2.51	75.6 ± 2.84	74.99 ± 2.72
ED-Polarity	59.48 ± 2.29	**61.93 ± 2.1** ∘	60.97 ± 1.96	61.73 ± 1.98 ∘	61.57 ± 2.04 ∘
STS-Neutrality	62.99 ± 2.03	**66.2 ± 2.11** ∘	65.26 ± 2.34 ∘	65.73 ± 1.98 ∘	65.77 ± 2.09 ∘
STS-PosNeg	**77.18 ± 2.88**	76.98 ± 2.82	75.39 ± 2.89 •	76.76 ± 2.89	76.98 ± 2.71
STS-Polarity	60.2 ± 2.23	**62.74 ± 1.52** ∘	62.34 ± 1.61 ∘	62.14 ± 1.73 ∘	62.1 ± 1.78 ∘
Weighted AUC					
Dataset	SO	ALL	SGD.TS+POS	SO.TS+POS	SO+POS
ED-Neutrality	0.62 ± 0.02	**0.65 ± 0.02** ∘	**0.65 ± 0.02** ∘	**0.65 ± 0.02** ∘	0.64 ± 0.02 ∘
ED-PosNeg	0.74 ± 0.03	**0.75 ± 0.03**	0.71 ± 0.03 •	0.74 ± 0.03	0.73 ± 0.03
ED-Polarity	0.62 ± 0.02	**0.65 ±0.02** ∘	0.64 ± 0.02	**0.65 ± 0.02** ∘	0.64 ± 0.02 ∘
STS-Neutrality	0.63 ± 0.02	**0.67 ± 0.02** ∘	0.66 ± 0.02 ∘	0.66 ± 0.02 ∘	0.66 ± 0.02∘
STS-PosNeg	**0.77 ± 0.03**	**0.77 ± 0.03**	0.75 ± 0.03 •	**0.77 ± 0.03**	**0.77 ± 0.03**
STS-Polarity	0.64 ± 0.02	**0.66 ± 0.01** ∘	0.65 ± 0.02 •	**0.66 ± 0.02** ∘	**0.66 ± 0.02** ∘

Table : World-level classification performance.

Expanded Lexicon

word	POS	label	negative	neutral	positive
alrighty	interjection	positive	0.021	0.087	0.892
boooooo	interjection	negative	0.984	0.013	0.003
lmaoo	interjection	positive	0.19	0.338	0.472
french	adjective	neutral	0.357	0.358	0.285
handsome	adjective	positive	0.007	0.026	0.968
saddest	adjective	negative	0.998	0.002	0
same	adjective	negative	0.604	0.195	0.201
anniversary	common.noun	neutral	0.074	0.586	0.339
tear	common.noun	negative	0.833	0.124	0.044
relaxing	verb	positive	0.064	0.244	0.692
wikipedia	proper.noun	neutral	0.102	0.644	0.254

Table : Expanded words example.

Expanded Lexicon (2)

(a) (b)

Figure : Word clouds of positive and negative words using log odds proportions.

Message-level classification

Accuracy				
Dataset	Baseline	ED	STS	Combination
6-coded	71.79 ± 2.79	$74.91 \pm 2.56 \circ$	$75.11 \pm 2.66 \circ$	$\mathbf{75.31 \pm 2.42} \circ$
Sanders	71.43 ± 3.76	$77.17 \pm 3.68 \circ$	$77.32 \pm 4.09 \circ$	$\mathbf{77.54 \pm 3.64} \circ$
SemEval	76.81 ± 1.22	76.66 ± 1.38	$77.7 \pm 1.25 \circ$	$\mathbf{78.13 \pm 1.38} \circ$
Weighted AUC				
Dataset	Baseline	ED	S140	Combination
6-coded	0.77 ± 0.03	$0.82 \pm 0.03 \circ$	$0.82 \pm 0.02 \circ$	$\mathbf{0.83 \pm 0.02} \circ$
Sanders	0.77 ± 0.04	$0.83 \pm 0.04 \circ$	$\mathbf{0.84 \pm 0.04} \circ$	$\mathbf{0.84 \pm 0.04} \circ$
SemEval	0.77 ± 0.02	$0.81 \pm 0.02 \circ$	$\mathbf{0.83 \pm 0.02} \circ$	$\mathbf{0.83 \pm 0.02} \circ$

Table : Message-level polarity classification performance.

Conclusions

- We presented a **supervised method** for opinion lexicon expansion on **Twitter**.
- The method creates a lexicon with **disambiguated** POS entries and a probability distribution for **positive, negative, and neutral** classes.
- Sentiment analysis methods that are based on **SentiWordnet** can be easily adapted to **Twitter** by relying on our lexicon.
- This method could be used to create **domain-specific** lexicons.
- It could also be used to study the **dynamics** of opinion-words.

Questions?

Thanks for your Attention!

Acknowledgments

- University of Waikato Doctoral Scholarship
- Machine Learning Group at the University of Waikato

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato