Supervised models for Twitter opinion lexicon expansion

Felipe Bravo-Marquez
Chief Supervisor: Bernhard Pfahringer
Supervisor: Eibe Frank

University of Waikato
Computer Science Department

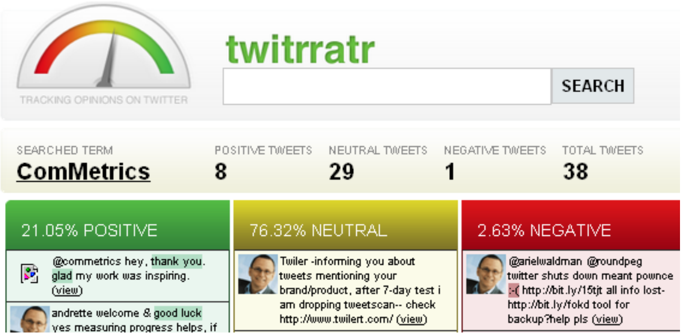
August 18, 2015

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.

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**.



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, wonderful** and **sad, bad**.

The Twitter dialect

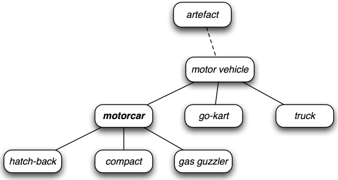
- The words used in Twitter include many abbreviations, acronyms, and misspelled words, e.g., **lol**, **omg**, **hahaha**, **#hatemonday**.
- This words are **not** covered by most popular lexicons.
- The manual creation of a Twitter-oriented opinion lexicon is a **time-consuming** task.

Proposal

- We study different **supervised models** for opinion lexicon expansion for **Twitter**.
- The proposed models create Twitter-oriented opinion lexicons in an automatic way.
- They rely on two resources: a **corpus of tweets** and a **seed lexicon**.
- Each expanded word has a **probability distribution**, describing how positive, negative, and neutral it is.

Previous work on lexicon expansion

- The expansion is normally done by exploiting **relations** between a **small seed lexicon** and **unknown words** from a **textual** resource.
- Two type of resources can be used: a lexical database such as **WordNet**, or a **corpus of documents**.



Lexicon expansion using WordNet

- Methods based on WordNet expand the seed words using semantic relations such **synonyms** and **antonyms**, [**?**, **?**].
- Hypothesis: synonyms have the **same** polarity and antonyms have the **opposite**.
- In [**?**] a **graph** was created using WordNet **adjectives** as vertices and the **synonym** relation as edges.
- Words are expanded by its **relative distance** from the two seed terms **good** and **bad**.
- In [**?**, **?**] the authors take the **dictionary definitions** of the seed words to train a word-level classifier.

Corpus-based lexicon expansion

- As semantic databases cover a fixed vocabulary, they **cannot** capture domain dependent words.
- Corpus approaches exploit **statistical patterns** observed in document corpora.
- They can potentially be applied to any domain, and hence, are more suitable for **Twitter lexicon expansion**.
- Hartziva et. al [**?**] used the conjunction relations between adjectives.
- Idea: adjectives connected with **and** tend to have the same polarity in the opposite way than adjectives connected with **but**.

Corpus-based lexicon expansion (2)

- Turney et.al proposed an unsupervised measure called **semantic orientation** (SO) [**?**] .
- It is calculated as the difference between the point-wise mutual information **PMI** of the word with a positive and a negative seed word.

$$\mathrm{PMI}(term_1, term_2) = \log_2\left(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1)Pr(term_2)}\right) \tag{1}$$

$$\mathrm{SO}(word) = \mathrm{PMI}(word, \text{``}excellent\text{''}) - \mathrm{PMI}(word, \text{``}poor\text{''}) \tag{2}$$

- The PMI values are estimated by the number of **hits** returned by a search engine.
- The resulting SO score is a numerical value whose **sign** represents the word's polarity.
- The magnitude of the value represents the **sentiment intensity**.

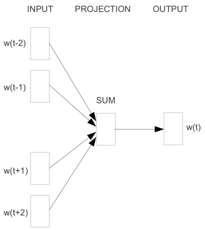
Twitter lexicon expansion

- Previous Twitter lexicon generation models compute the SO between **corpus words** and tweet-level sentiment **labels**.
- The tweets are **automatically** labelled to polarity classes using **distant supervision** [**?**, **?**] or **self-training** [**?**].
- Distant supervision methods rely on strong **sentiment clues** found in the message such as **emoticons** [**?**, **?**] or **hashtags** [**?**] to label the messages.
- Tweets where these clues are not observed are **discarded**.

positive	negative
:)	:(
:-)	:-(
:D	=(
=)	:'(

Twitter lexicon expansion using word embeddings

- Word **embeddings** are low-dimensional continuous dense word vectors trained from document corpora.
- Most popular models are skip-gram [**?**], continuos bag-of-words [**?**], and Glove [**?**].



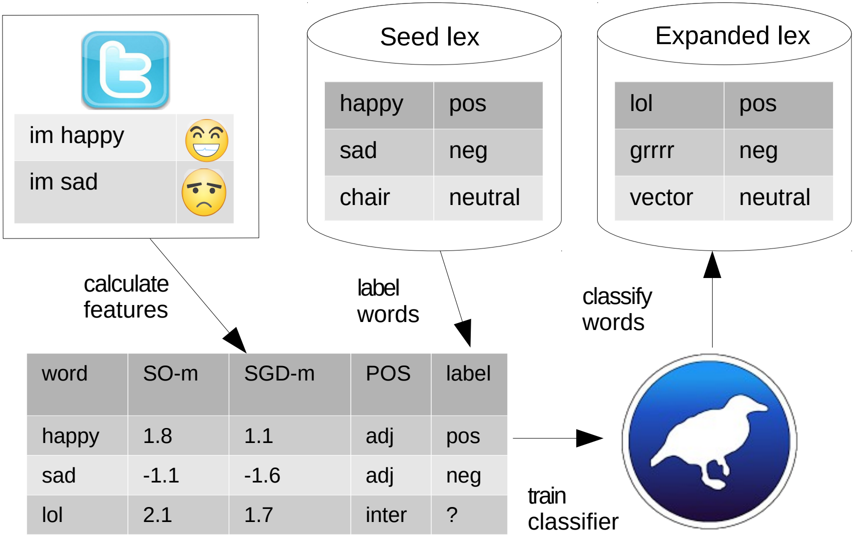
CBOW

- In [**?**], they were used as **features** in a regression model for determining the association between Twitter words and **positive sentiment**.
- In [**?**] **sentiment-specific** word embeddings are proposed by combining the skip-gram model with emoticon-annotated tweets.
- These embeddings were used for **training** a word-level polarity classifier.

A model based on labelled tweets

- We propose a **supervised framework** for **Twitter** lexicon expansion based on **sentiment-annotated tweets**.
- The tweets are labelled in an **automatic fashion** using emoticons.
- 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.
- These properties are inspired by **SentiWordnet**.

Methodology



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
Meta-Lex	3730	6368	7088

Table : Lexicon Statistics

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 Features

- 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.

$$\frac{\lambda}{2}||w||^2 + \sum [1 - y(\mathbf{xw} + b)]_+. \qquad (3)$$

- 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 presence of a word **influences** the prediction of **polarity** classes.
- We use **time windows** of $1,000$ examples.

The SO time-series

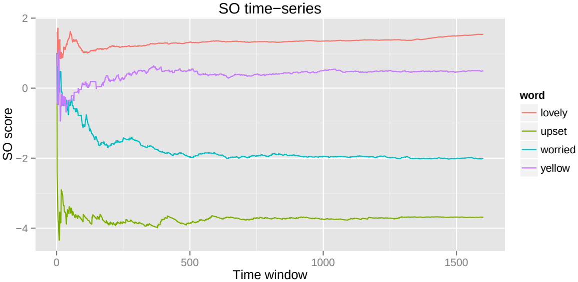
- The second time-series corresponds to the **accumulated semantic orientation** (SO).
- It is based on the **point-wise mutual information** measure.

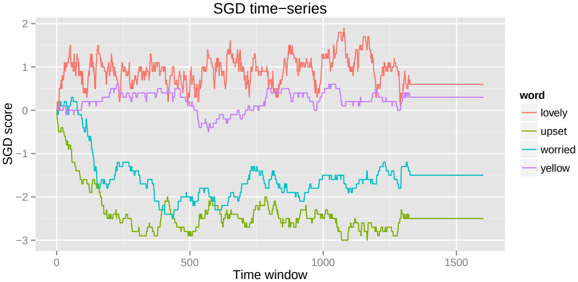
$$\begin{aligned}
\mathrm{SO}(w) &= \mathrm{PMI}(w, \mathrm{pos}) - \mathrm{PMI}(w, \mathrm{neg}) \\
&= \log_2\left(\frac{\Pr(w, pos)}{\Pr(w) \times \Pr(pos)}\right) - \log_2\left(\frac{\Pr(w, neg)}{\Pr(w) \times \Pr(neg)}\right) \\
&= \log_2\left(\frac{\Pr(w, pos) \times \Pr(neg)}{\Pr(pos) \times \Pr(w, neg)}\right)
\end{aligned} \tag{4}$$

$$\mathrm{SO}(w) = log_2\left(\frac{\mathrm{count}(w \wedge y = 1) \times \mathrm{count}(y = -1)}{\mathrm{count}(w \wedge y = -1) \times \mathrm{count}(y = 1)}\right) \tag{5}$$

- 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.trunc.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.trunc.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

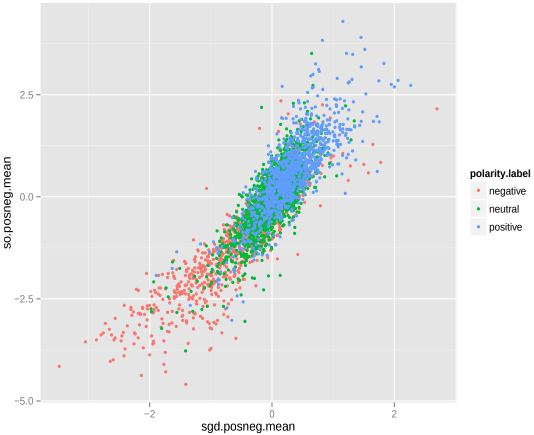
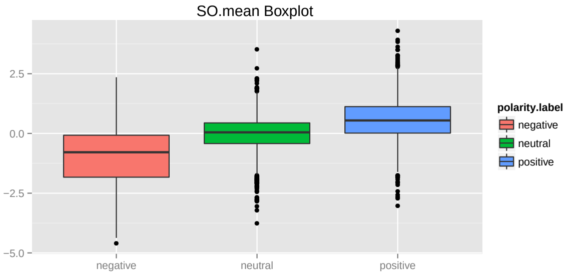
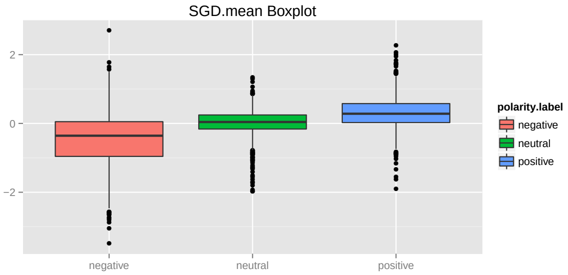


Figure : SO vs SGD scatterplot.

Feature Visualisation (2)





Word-level Classification Results using RBF SVMs

Weighted AUC		
Dataset	SO	ALL
ED-Polarity	0.62 ± 0.02	**0.65** ±**0.02** ∘
STS-Polarity	0.64 ± 0.02	**0.66** ± **0.01** ∘
Kappa		
Dataset	SO	ALL
ED-Polarity	0.28 ± 0.04	**0.33** ±**0.04** ∘
STS-Polarity	0.31 ± 0.04	**0.35** ± **0.03** ∘

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

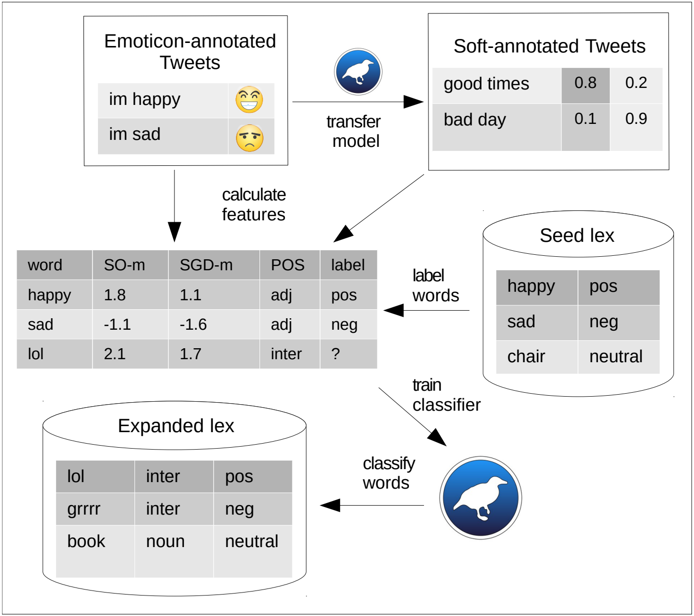
Accuracy				
Dataset	Baseline	ED	STS	Combination
6-coded	71.79 $\pm$ 2.79	74.91 $\pm$ 2.56 $\circ$	75.11 $\pm$ 2.66 $\circ$	**75.31 $\pm$ 2.42** $\circ$
Sanders	71.43 $\pm$ 3.76	77.17 $\pm$ 3.68 $\circ$	77.32 $\pm$ 4.09 $\circ$	**77.54 $\pm$ 3.64** $\circ$
SemEval	76.81 $\pm$ 1.22	76.66 $\pm$ 1.38	77.7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

Table : Message-level polarity classification performance.

Discussions

- 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.
- This method depends on a collection of **emoticon-annotated tweets**.
- It would be hard to apply to **domains** where emoticons are not **frequently used**.

Information Fusion Extension



Lexicon generation from unlabelled Tweets

- We propose another supervised model for lexicon expansion referred to as the **tweet-centroid model**.
- The words are represented by **high-dimensional vectors** based on the context's where they occur.
- In contrast to the previous approach the expansion is done from **unlabelled tweets**.
- It is inspired by the **Distributional Hypothesis** [**?**]: words occurring in the same **contexts** tend to have similar meanings.
- Or equivalently: "a word is characterized by the **company** it keeps".

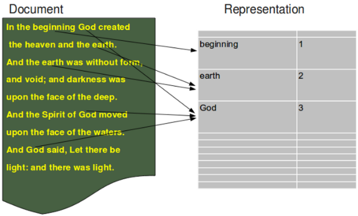
Lexicon generation from unlabelled Tweets (2)

- We treat a **whole tweet** as a word's context.
- We model tweets as **vectors** calculated from the **content**.
- We calculate word-level vectors based on the **centroids** of the **tweet-vectors** where a word occurs.
- We suggest that words exhibiting a **certain polarity** are more likely used in contexts expressing the **same polarity** than in contexts exhibiting a **different one**.

Lexicon generation from unlabelled Tweets (3)

Tweets in the collection are represented by **two vectors**:

1. A word-frequency **bag-of-words vector**.
2. A semantic vector based on **word-clusters**.



Bag-of-words model

- Suppose we have a **corpus** $\mathcal{C}$ formed by n tweets $t_1, \ldots, t_n$.
- Each **tweet** t is a sequence of words.
- Let $\mathcal{V}$ be the **vocabulary** formed by the m different words $w_1, \ldots, w_m$ found in $\mathcal{C}$.
- The tweet-level bag-of-words model represents each tweet t as a **m-dimensional vector** $\overrightarrow{t}$.
- Each dimension $\overrightarrow{t}_j$ corresponds to the **frequency** in which the word w_j appears in t.

Bag-of-words model (2)

- We define the **word-tweet set** $\mathcal{W}(w)$ as the set of tweets in which w is **observed**:

$$\mathcal{W}(w) = \{t : w \in t, \forall t \in \mathcal{C}\} \tag{6}$$

- We define the word-level vector $\overrightarrow{w}$ as as the **centroid** of all tweet-vectors in which w is used.
- $\overrightarrow{w}$ is a m-dimensional vector in which each dimension w_j is calculated as follows:

$$\overrightarrow{w}_j = \sum_{t \in \mathcal{W}(w)} \frac{f_j(t)}{|\mathcal{W}(w)|} \tag{7}$$

- Bag-of-words models tend to produce **high dimensional sparse vectors**.

Word-clusters model

- Let be c a **clustering** functions that **maps** the m words to a **partition** of the vocabulary $\mathcal{S}$ of k classes, with $k \ll m$.
- This function is trained in an **unsupervised fashion** from a corpus of tweets using the **Brown clustering** algorithm [**?**].
- This algorithm produces **hierarchical clusters** by maximising the mutual information of **bigrams**.
- These clusters have shown to be useful for tagging tweets according to **part-of-speech** classes [**?**].

Word-clusters model (2)

- We **tag** the word sequences of the tweets from $\mathcal{C}$ with the clustering function c.
- We create a new **tweet-level** vector $\overrightarrow{tc}$ of k dimensions based on the **frequency** of occurrence of a cluster s in the tweet.
- We take the **centroids** of the cluster-based vectors $\overrightarrow{tc}$ from the tweets of $\mathcal{W}(w)$, producing k-dimensional word vectors.

Tweet-centroid Model

vocabulary

w1	angry
w2	happy
w3	good
w4	grr
w5	lol

tweets

lol happy
lol good
grr angry

word clusters

w1	c1
w2	c2
w3	c2
w4	c1
w5	c2

$\overrightarrow{tb}$

	w1	w2	w3	w4	w5
t1	0	1	0	0	1
t2	0	0	1	0	1
t3	1	0	0	1	0

$\overrightarrow{tc}$

	c1	c2
t1	0	2
t2	0	2
t3	2	0

$\overrightarrow{wb}$

	w1	w2	w3	w4	w5
w1	1	0	0	1	0
w2	0	1	0	0	1
w3	0	0	1	0	1
w4	1	0	0	1	0
w5	0	0.5	0.5	0	1

$\overrightarrow{wc}$

	c1	c2
w1	2	0
w2	0	2
w3	0	2
w4	2	0
w5	0	2

Generated lex

lol	pos
grr	neg

Seed lex

happy	pos
angry	neg
good	pos

Datasets

- We take a **random sample** of 2.5 million English tweets from the Edinburgh Corpus and STS.
- ED corpus represents a **realistic sample** from a stream of tweets.
- STS was **intentionally manipulated** to over-represent the presence of subjective tweets.
- We study these datasets to observe the effects of **manipulating** the collection of tweets for lexicon generation.
- We tokenise the tweets from both collections and create the word-level vectors $\overrightarrow{w}$ and $\overrightarrow{wc}$.

Dataset	STS	ED
#tweets	1,600,000	2,500,000
#positive words	2015	2639
#negative words	2621	3642
#neutral words	3935	5085
#unlabelled words	36,451	67,692
#bag-of-words attributes	45,022	79,058
#cluster-vector attributes	993	999

Word-level 2-class polarity classification performance

Accuracy			
Dataset	WORDS	CLUSTER	ALL
STS	75.52 ± 1.81	$77.2 \pm 1.9 \circ$	$\mathbf{77.85 \pm 1.94} \circ$
ED	77.75 ± 1.54	77.62 ± 1.37	$\mathbf{79.15 \pm 1.39} \circ$
AUC			
Dataset	WORDS	CLUSTER	ALL
STS	0.83 ± 0.02	$0.84 \pm 0.02 \circ$	$\mathbf{0.85 \pm 0.02} \circ$
ED	0.85 ± 0.01	0.85 ± 0.01	$\mathbf{0.86 \pm 0.01} \circ$

Word-level 3-class polarity classification performance

Accuracy			
Dataset	WORDS	CLUSTER	ALL
STS	61.84 ± 1.46	64.42 ± 1.54 ◦	$\mathbf{64.57} \pm \mathbf{1.44}$ ◦
ED	62.93 ± 1.31	64.5 ± 1.16 ◦	$\mathbf{65.5} \pm \mathbf{1.19}$ ◦

AUC			
Dataset	WORDS	CLUSTER	ALL
STS	0.77 ± 0.01	$\mathbf{0.79} \pm \mathbf{0.01}$ ◦	$\mathbf{0.79} \pm \mathbf{0.01}$ ◦
ED	0.78 ± 0.01	0.79 ± 0.01 ◦	$\mathbf{0.8} \pm \mathbf{0.01}$ ◦

Generated words example

word	label	negative	neutral	positive
#recession	negative	0.603	0.355	0.042
#silicon_valley	neutral	0.043	0.609	0.348
bestfriends	positive	0.225	0.298	0.477
christamas	positive	0.003	0.245	0.751
comercials	negative	0.678	0.317	0.005
hhahaha	positive	0.112	0.409	0.479
powerpoint	neutral	0.068	0.802	0.13
psychotic	negative	0.838	0.138	0.024
widows	negative	0.464	0.261	0.275
yassss	positive	0.396	0.08	0.524

Message-level classification performance

Accuracy			
Dataset	Baseline	STS	ED
Sanders	73.25 ± 3.51	74.76 ± 4.21	$\mathbf{76.58 \pm 3.8}$ $\circ$
6-human	72.84 ± 2.57	75.08 ± 2.31 $\circ$	$\mathbf{76.42 \pm 2.34}$ $\circ$
SemEval	77.72 ± 1.24	78.97 ± 1.31 $\circ$	$\mathbf{79.18 \pm 1.22}$ $\circ$

AUC			
Dataset	Baseline	STS	ED
Sanders	0.78 ± 0.04	0.8 ± 0.04 $\circ$	$\mathbf{0.83 \pm 0.04}$ $\circ$
6-human	0.79 ± 0.03	0.82 ± 0.03 $\circ$	$\mathbf{0.83 \pm 0.02}$ $\circ$
SemEval	0.78 ± 0.02	0.82 ± 0.02 $\circ$	$\mathbf{0.84 \pm 0.02}$ $\circ$

Discussions

- Collections of tweets manipulated to over-represent subjective tweets are not necessarily **better** for lexicon generation than random collections of tweets.
- The proposed technique relies on resources that are relatively **cheap** to obtain: a **seed lexicon**, and a collection of **unlabelled tweets**.
- Source code available:
  `http://www.cs.waikato.ac.nz/ml/sa/lex.html`.

Comparison of both models

	Time-series	Tweet-Centroid
Depends on a seed lexicon	Yes	Yes
Detects neutral words	Yes	Yes
Depends on labelled tweets	Yes	No
Dimensionality	Low	High
POS disambiguation	Yes	No
Considers time	Yes	No
Suitable for stream learning	Yes	Yes
Suitable for emotion detection	No	Yes

Future Work

- Use the Tweet-centroid model for **emotion** classification using **multi-label classification**.
- Transfer other message-level attributes to the word-level, e.g., n-grams, POS-tags.
- Design a mechanism to discover opinion words in an **online fashion**.

Questions?

Thanks for your Attention!

Acknowledgments

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

References I

Amir, S., Ling, W., Astudillo, R., Martins, B., Silva, M. J., and Trancoso, I. (2015).
Inesc-id: A regression model for large scale twitter sentiment lexicon induction.
In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 613–618, Denver, Colorado. Association for Computational Linguistics.

Becker, L., Erhart, G., Skiba, D., and Matula, V. (2013).
Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion.
Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992).
Class-based n-gram models of natural language.
Computational linguistics, 18(4):467–479.

Esuli, A. and Sebastiani, F. (2005).
Determining the semantic orientation of terms through gloss classification.
In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 617–624, New York, NY, USA. ACM.

References II

Esuli, A. and Sebastiani, F. (2006).
Sentiwordnet: A publicly available lexical resource for opinion mining.
In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06*, pages 417–422.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011).
Part-of-speech tagging for twitter: Annotation, features, and experiments.
In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Harris, Z. (1954).
Distributional structure.
Word, 10(23):146–162.

Hatzivassiloglou, V. and McKeown, K. R. (1997).
Predicting the semantic orientation of adjectives.
In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

References III

Hu, M. and Liu, B. (2004).
Mining and summarizing customer reviews.
In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Kamps, J., Marx, M., Mokken, R. J., and De Rijke, M. (2004).
Using WordNet to Measure Semantic Orientation of Adjectives.
In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 4, pages 1115–1118. European Language Resources Association (ELRA).

Kim, S.-M. and Hovy, E. (2004).
Determining the sentiment of opinions.
In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).
Distributed representations of words and phrases and their compositionality.
In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

References IV

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013).
Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets.
Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013).

Pennington, J., Socher, R., and Manning, C. D. (2014).
Glove: Global vectors for word representation.
In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014).
Building large-scale twitter-specific sentiment lexicon : A representation learning approach.
In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 172–182.

Turney, P. D. and Littman, M. L. (2003).
Measuring praise and criticism: Inference of semantic orientation from association.
ACM Transactions on Information Systems (TOIS), 21(4):315–346.

References V

Zhou, Z., Zhang, X., and Sanderson, M. (2014).
Sentiment analysis on twitter through topic-based lexicon expansion.
In Wang, H. and Sharaf, M., editors, *Databases Theory and Applications*, volume 8506 of *Lecture Notes in Computer Science*, pages 98–109. Springer International Publishing.