Analysing Sentiment and Emotions in Social Media

Felipe Bravo-Marquez

Department of Computer Science, University of Chile

May 8, 2019

Message-level Polarity Classification (MPC)

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

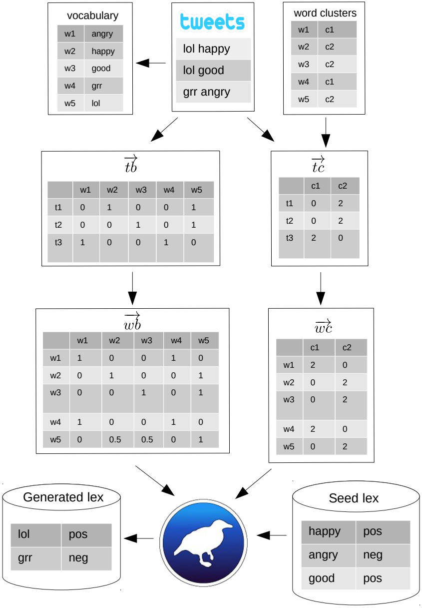
2. Challenge: Tweets use a unique **informal dialect** including many abbreviations, acronyms, misspelled words, hashtags, and emoticons, e.g., **lol**, **omg**, **hahaha**, **#hatemonday**, **#SweetAsBro**, **#yeahnah**, **:)** .
3. State-of-the-art solutions use **supervised** machine learning models trained from **manually** annotated examples [Kiritchenko et al., 2014].
4. **Label sparsity problem (LS)**: manual annotation is **labour-intensive** and **time-consuming**.

Research Problem

The models presented in this talk address the label sparsity problem for Twitter sentiment classification by automatically building **two type of resources**.

1. **Twitter-specific opinion lexicons**: we develop machine learning models to induce polarity lexicons from tweets.
2. **Synthetically labelled tweets**: we develop distant supervision methods based on **lexical knowledge** (we go beyond emoticons).

Tweet-centroid Model for Lexicon Induction

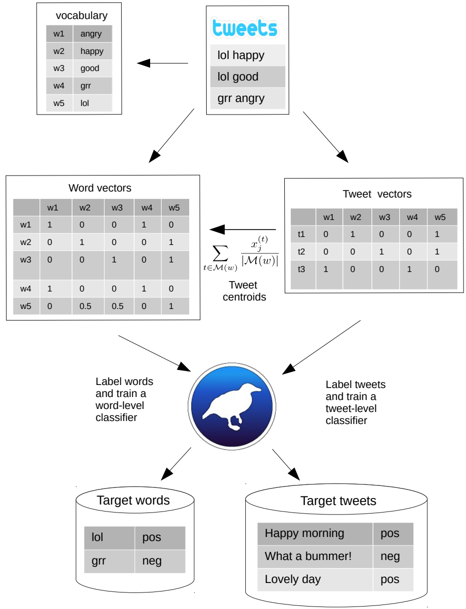


Message-level classification performance

AUC			
Dataset	Baseline	STS	ED
Sanders	0.78 ± 0.04	$0.80 \pm 0.04 +$	**0.83** $\pm$ **0.04** $+$
6-human	0.79 ± 0.03	$0.82 \pm 0.03 +$	**0.83** $\pm$ **0.02** $+$
SemEval	0.78 ± 0.02	$0.82 \pm 0.02 +$	**0.84** $\pm$ **0.02** $+$

Multi-Label Classification of Emotions with TCM

Transfer Learning with Tweet Centroids



Lexicon Induction with Transfer Learning

- What if we don't have a seed lexicon?
- We can train a **message-level classifier** f_M from a corpus of sentiment annotated tweets $\mathcal{C}_L$ and deploy it on words found in a **corpus of unlabelled tweets** represented by tweet centroids.
- Tweets are represented by **sparse vectors** using unigrams, Brown clusters, and POS tags.
- Note that tweets and words reside in the **same feature space**.

	AUC	
Source Dataset	PMI-SO	TCM
Sanders	0.757	**0.864**
6HumanCoded	0.861	**0.930**
SemEval	0.858	**0.916**

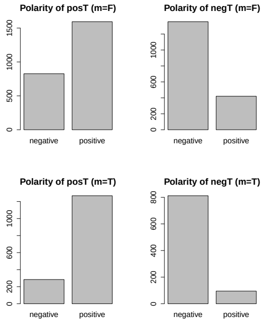
Table: Word-level Polarity Classification Results for the AFINN lexicon.

Lexical-based Distant Supervision

- Lexicons showed to be **useful features** for MPC.
- But we **still need labelled tweets** for training a message-level classifier.
- We will try to **directly use** lexical knowledge for training message-level classifiers.
- We propose two **distant supervision** models: **Partitioned Tweet Centroids** and **Annotate-Sample-Average (ASA)**.
- Proposed methods generate positive and negative training instances by **averaging** tweets containing words with the **same** polarity.

Lexical Polarity Hypothesis

- A tweet containing a word with a certain polarity is more likely to express the **same polarity** than the **opposite** $p_d > 0.5$ (Bernoulli experiment).



- The opposite polarity may also be expressed due to the presence of **negation**, **sarcasm**, or other opinion words with the **opposite** polarity.

Why Averaging?

- Averaging multiple tweets with words with the same polarity **increases** the confidence of generating instances located in the **region** of the desired polarity.
- We assume that the average tweet will behave similarly to the **majority**.
- Probability that the **majority** of the tweets sampled from a collection of tweets with at least one word with the target polarity have the desired polarity:

$$P(M) = \sum_{i=\lfloor \frac{a}{2} \rfloor + 1}^{a} \binom{a}{i} p_d^i (1 - p_d)^{a-i}$$

	$p_d = 0.6$	$p_d = 0.7$	$p_d = 0.8$	$p_d = 0.9$
$a = 3$	0.648	0.784	0.896	0.972
$a = 5$	0.683	0.837	0.942	0.991
$a = 10$	0.633	0.850	0.967	0.998
$a = 50$	0.902	0.998	1	1
$a = 100$	0.973	1	1	1
$a = 500$	1	1	1	1
$a = 1000$	1	1	1	1

- $P(M) > p_d$, when $a \geq 3$ and $p_d \geq 0.5$. This is analogous to the **Condorcet's Jury Theorem**!!

TCM for message-level classification

- TCM can be used as a **distant supervision** model for MPC.
- We use a **word-level** classifier f_W trained with TCM vectors calculated from $\mathcal{C}_U$ labelled by a **polarity lexicon** $\mathcal{L}$ (AFINN).
- The classifier is deployed on the target tweets represented by **sparse vectors**.
- The number of labelled words for training f_W is **limited** to the number of words from $\mathcal{L}$.
- TCM is **not capable** of exploiting large collections of unlabelled tweets for producing training datasets larger than the size of $\mathcal{L}$.

Partitioned TCM

- We propose a modification of our method for **increasing** the number labelled instances it produces.
- The word-tweet set $\mathcal{M}(w)$ for each word from the lexicon ($w \in \mathcal{L}$) is **partitioned** into smaller disjoint subsets $\mathcal{M}(w)_1, \dots \mathcal{M}(w)_z$ of a fixed size determined by a parameter p.
- We calculate one tweet centroid vector $\overrightarrow{w}$ for **each partition** labelled according to $\mathcal{L}$.

Baselines

Emoticon-Annotation Approach (EAA)

- Labels tweets with positive or negative emoticons according to the emoticon's polarity after removing the emoticon from the message.
- Tweets containing both positive and negative emoticons are **discarded**.

Lexicon-annotation approach (LAA)

- Uses a given polarity lexicon $\mathcal{L}$.
- Tweets with at least one positive word and no negative word are labelled **positive**.
- Tweets with at least one negative word and no positive word are labelled **negative**.

TCM for MPC

	6HumanCoded		Sanders		SemEval	
EAA	0.805 $\pm$ 0.005	= -	0.800 $\pm$ 0.017	= +	0.802 $\pm$ 0.006	= -
LAA	0.809 $\pm$ 0.001	+ =	0.778 $\pm$ 0.002	- =	0.814 $\pm$ 0.000	+ =
TCM	0.776 $\pm$ 0.004	- -	0.682 $\pm$ 0.024	- -	0.779 $\pm$ 0.008	- -
TCM (p=5)	0.834 $\pm$ 0.002	+ +	0.807 $\pm$ 0.008	= +	0.833 $\pm$ 0.002	+ +
TCM (p=10)	0.845 $\pm$ 0.003	+ +	**0.817** $\pm$ 0.006	+ +	0.841 $\pm$ 0.002	+ +
TCM (p=20)	**0.850** $\pm$ 0.003	+ +	0.815 $\pm$ 0.011	+ +	**0.844** $\pm$ 0.003	+ +
TCM (p=50)	0.844 $\pm$ 0.004	+ +	0.785 $\pm$ 0.010	- +	0.836 $\pm$ 0.004	+ +
TCM (p=100)	0.829 $\pm$ 0.003	+ +	0.752 $\pm$ 0.019	- -	0.821 $\pm$ 0.004	+ +

Table: Message-level Polarity Classification Results. Best results per column are given in bold.

Annotate-Sample-Average (ASA)

- Partitioned TCM can generate **very large** training datasets.
- TCM instances are obtained by averaging tweets containing **the same word**.
- What if we average random tweets containing **different words** with the same polarity?
- What if we can define the **number of instances** to generate?
- This could be useful for creating **compact and balanced** training datasets.

Annotate-Sample-Average (ASA)

- **Annotation**: every time a word from $\mathcal{L}$ is found, the tweet is added to sets **posT** or **negT** (depending on the polarity).
- **Sample**: randomly sample with replacement *a* tweets from either **posT** or **negT** for each generated instance.
- **Averaging**: average and label sampled feature vectors.
- We create balanced training datasets with size equal to 1% of the size of the source corpus ($20,000$ in our experiments).

ASA results

	6HumanCoded		Sanders		SemEval	
EAA_U	0.805 ± 0.005	= = - -	0.800 ± 0.017	= = + +	0.802 ± 0.006	= + - -
EAA_B	0.809 ± 0.001	= = = =	0.795 ± 0.016	= = + +	0.798 ± 0.007	- = - -
LAA_U	0.809 ± 0.001	+ = = =	0.778 ± 0.002	- - = =	0.814 ± 0.000	+ + = =
LAA_B	0.809 ± 0.001	+ = = =	0.778 ± 0.003	- - = =	0.813 ± 0.001	+ + = =
ASA ($a = 1$, $m = F$)	0.793 ± 0.005	- - - -	0.762 ± 0.016	- - - -	0.787 ± 0.007	- - - -
ASA ($a = 5$, $m = F$)	0.837 ± 0.004	+ + + +	0.807 ± 0.010	= = + +	0.833 ± 0.003	+ + + +
ASA ($a = 10$, $m = F$)	$\mathbf{0.845} \pm 0.001$	+ + + +	$\mathbf{0.812} \pm 0.015$	+ + + +	$\mathbf{0.840} \pm 0.003$	+ + + +
ASA ($a = 50$, $m = F$)	0.815 ± 0.003	+ + + +	0.759 ± 0.006	- - - -	0.810 ± 0.004	+ + - -
ASA ($a = 100$, $m = F$)	0.781 ± 0.003	- - - -	0.720 ± 0.007	- - - -	0.779 ± 0.004	- - - -
ASA ($a = 500$, $m = F$)	0.723 ± 0.002	- - - -	0.670 ± 0.008	- - - -	0.729 ± 0.005	- - - -
ASA ($a = 1000$, $m = F$)	0.712 ± 0.002	- - - -	0.665 ± 0.007	- - - -	0.721 ± 0.005	- - - -

Table: AUC measure for different distant supervision models. Best results per column are given in bold.

Conclusions

- The methods presented in this talk can be used to **acquire** and **exploit** lexical knowledge for Twitter sentiment analysis under **label sparsity conditions**.
- We proposed two methods (Word Sentiment Associations and TCM) for building Twitter-specific **opinion lexicons** (acquisition of lexical knowledge).
- These methods could be used to create **domain-specific** lexicons.
- They could also be used to study the **dynamics** of opinion-words.
- Future work: try **non-linear representations** on TCM (Auto-Encoders or RBM).

Other projects

- WASSA 2017 Shared Task in Emotion Intensity: given a tweet an emotion X (anger, fear, joy, or sadness) determine the intensity or degree of emotion X felt by the speaker—a real-valued score between 0 and 1.
- English tweets were annotated using Best-Worst scaling.
- Twenty-two teams participated. Best system: ensemble of deep learning models ($r = 0.74$).
- SemEval 2018 Task 1: Affect in Tweets. Extension of previous task including VAD emotions and two more languages: Spanish and Arabic.
- AffectiveTweets Weka Package

Questions?

Thanks for your Attention!

References I

Kiritchenko, S., Zhu, X., and Mohammad, S. M. (2014).
Sentiment analysis of short informal texts.
Journal of Artificial Intelligence Research, 50:723–762.